

Segment Routing Configuration Guide, Cisco IOS XE 17 | Access and Edge Routers

First Published: 2020-04-28

Last Modified: 2024-08-26

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

© 2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1 **Read Me First** 1

Short Description 2

CHAPTER 2 **Introduction to Segment Routing for the MPLS Dataplane** 3

Feature Information for Segment Routing 3

Overview of Segment Routing for the MPLS Dataplane 3

How Segment Routing Works 4

Examples for Segment Routing 5

Benefits of Segment Routing 6

Segment Routing Global Block 8

Adjacency Segment Identifiers 9

Prefix Segment Identifiers 9

Additional References for Segment Routing 9

CHAPTER 3 **Segment Routing with IS-IS v4 Node SID** 11

Feature Information for Segment Routing with IS-IS v4 Node SID 11

Restrictions for Segment Routing with IS-IS v4 Node SID 11

Information About Segment Routing with IS-IS v4 Node SID 12

Segment Routing IS-IS v4 Node SID 12

Prefix SIDs Received in Label-Switched Path from Remote Routers 12

Segment Routing Adjacency SID Advertisement 13

Adjacency SIDs 13

Segment Routing Mapping Server 13

Connected Prefix SIDs 14

SRGB Range Changes 14

SRGB Deletion 14

MPLS Forwarding on an Interface 14
Segment Routing and LDP Preference 14
Segment Routing Traffic Engineering Announcements 15
How to Configure Segment Routing with IS-IS v4 Node SID 15
Configuring Segment Routing 15
Configuring Segment Routing on an IS-IS Network 16
Configuring Prefix-SID for IS-IS 17
Configuring Prefix Attribute N-Flag 19
Configuring the Explicit Null Attribute 20
Configuring Segment Routing Label Distribution Protocol Preference 21
Configuring IS-IS SRMS 22
Configuring IS-IS SRMS Client 22
Configuring IS-IS SID Binding TLV Domain Flooding 22
Configuration Examples for Segment Routing —IS-IS v4 Node SID 23
Example: Configuring Segment Routing on IS-IS Network 23
Example: Configuring an Explicit Null Attribute 23
Additional References for Segment Routing with IS-IS v4 Node SID 23

CHAPTER 4 **IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute 25**
Feature Information for IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute 25
Prerequisites for IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute 26
Restrictions for IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute 27
Information About IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute 27
Topology-Independent Loop Free Alternate 28
Topology Independent Loop Free Alternate Tie-break 28
Interface Fast Reroute Tiebreakers 29
How to Configure IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute 30
Configuring Topology Independent Loop Free Alternate Fast Reroute 30
Configuring Topology Independent Loop Free Alternate With Mapping Server 30
Examples: Configuring IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute 34
Verifying the Tiebreaker 35
Verifying the Primary and Repair Paths 35

Verifying the IS-IS Segment Routing Configuration 36
Verifying the IS-IS Topology Independent Loop Free Alternate Tunnels 37
Verifying the Segment Routing Traffic Engineering With Topology Independent Loop Free Alternate Configuration 38

CHAPTER 5 **Segment Routing Traffic Engineering With IS-IS 41**
Feature Information for Segment Routing Traffic Engineering with IS-IS 41
Restrictions for Segment Routing-Traffic Engineering with IS-IS 42
Information About Segment Routing Traffic Engineering with IS-IS 42
SR-TE LSP Instantiation 42
SR-TE LSP Explicit Null 43
SR-TE LSP Path Verification 43
SR-TE Traffic Load Balancing 45
SR-TE Tunnel Reoptimization 45
SR-TE with Lockdown Option 46
SR-TE Tunnel Protection 47
Unnumbered Support 48
How to Configure Segment Routing Traffic Engineering with IS-IS 48
Configuring the Path Option for a TE Tunnel 48
Configuring SR Explicit Path Hops 48
Configuring Affinity on an Interface 49
Use Case: Segment Routing Traffic Engineering Basic Configuration 49
Explicit Path SR-TE Tunnel 1 51
Explicit Path SR-TE Tunnel 2 51
Explicit Path SR-TE Tunnel 3 52
Dynamic Path SR-TE Tunnel 4 52
Dynamic Path SR-TE Tunnel 5 52
Verifying Configuration of the SR-TE Tunnels 53
Verifying Tunnel 1 53
Verifying Tunnel 2 53
Verifying Tunnel 3 54
Verifying Tunnel 4 54
Verifying Tunnel 5 55
Verifying Verbatim Path Support 56

CHAPTER 6 **Segment Routing With OSPFv2 Node SID** 57

Feature Information for Segment Routing With OSPFv2 Node SID 57

Information About Segment Routing With OSPFv2 Node SID 57

Prefix-SID Received in Label Switched Path From Remote Routers 58

Segment Routing Adjacency SID Advertisement 58

Multiple Adjacency-SIDs 59

Segment Routing Mapping Server 59

Connected Prefix SIDs 59

SRGB Range Changes 59

MPLS Forwarding on an Interface 59

Conflict Handling of SID Entries 60

How to Configure Segment Routing With OSPFv2 Node SID 60

Configuring Segment Routing With OSPF 60

Configuring Segment Routing on OSPF Network 61

Configuring Prefix-SID for OSPF 63

Configuring Prefix Attribute N-flag-clear 64

Configuring Explicit Null Attribute With OSPF 65

Configuring Segment Routing Label Distribution Protocol Preference With OSPF 66

Configuring OSPF SRMS 67

Configuring OSPF SRMS Client 68

Additional References for Segment Routing With OSPFv2 Node SID 68

CHAPTER 7 **OSPFv2 Link-protection Topology Independent Loop Free Alternate Fast Reroute** 69

Feature Information for OSPFv2 Link-protection Topology Independent Loop Free Alternate Fast Reroute 69

Restrictions for Topology Independent Loop Free Alternate Fast Reroute 70

Information About OSPFv2 Link-Protection Topology Independent Loop Free Alternate Fast Reroute 71

IP Fast Reroute and Remote Loop Free Alternate 71

Topology Independent Fast Reroute 71

Topology-Independent Loop Free Alternate 72

Topology Independent Loop Free Alternate Tie-break 72

P-Space 73

Q-Space 73
Post-Convergence Path 73
Per-Destination Link Protection 74
Per Interface Loop Free Alternate Enablement 75
Prefix Processing 75
Anycast Prefix Processing 75
Per-Prefix Loop Free Alternate Tie-Break 75
Node Protection 77
Shared Risk Link Groups Protection 77
Node-Shared Risk Link Groups Protection 78
How to Configure Topology Independent Loop Free Alternate Fast Reroute 78
Enabling Topology Independent Loop Free Alternate Fast Reroute 78
Configuring Topology Independent Loop Free Alternate Fast Reroute 79
Configuring Topology Independent Fast Reroute Tie-breaker 80
Verifying Topology Independent Fast Reroute Tunnels 81
Debugging Topology Independent Loop Free Alternate Fast Reroute 83
Examples: OSPFv2 Link-Protection Topology Independent Loop Free Alternate Fast Reroute 83
Example: Configuring Topology Independent Loop Free Alternate Fast Reroute 83

CHAPTER 8 **Segment Routing Traffic Engineering With OSPF 85**
Feature Information for Segment Routing Traffic Engineering With OSPF 85
Restrictions for Segment Routing Traffic Engineering With OSPF 86
Information About Segment Routing Traffic Engineering With OSPF 86
Benefits of Using Segment Routing Traffic Engineering With OSPF 86
OSPFv2 Segment Routing Traffic Engineering Functionalities 87
Protected Adjacency SID 87
Traffic Engineering Interfaces 87
Unnumbered Support 87
Segment Routing Traffic Engineering Support for Forwarding Adjacency 87
Segment Routing Traffic Engineering Support for Auto-route Announce 88
Auto-route Announce IP2MPLS 88
SR-TE LSP Instantiation 88
Tunnel Path Affinity Validation 88
SR-TE Traffic Load Balancing 89

Load Balancing on Port Channel TE Links 89
Load Balancing on Single Tunnel 89
Load Balancing on Multiple Tunnels 89
SR-TE Tunnel Reoptimization 89
SR-TE with Lockdown Option 90
SR-TE Tunnel Protection 91
IP-FRR Local Repair Protection 91
Tunnel Path Protection 91
SR-TE LSP Path Verification 92
Topology Path Validation 92
SR SID Validation 92
LSP Egress Interface 93
IP Reachability Validation 93
Tunnel Path Resource Avoidance Validation 93
SR-TE LSP Explicit Null 93
Verbatim Path Support 94
How to Configure Segment Routing Traffic Engineering With OSPF 94
Enabling Segment Routing Traffic Engineering With OSPF 94
Configuring the Path Option for a TE Tunnel 94
Configuring SR Explicit Path Hops 94
Configuring Tunnel Path Affinity Validation 95
Configuring Affinity on an Interface 96
Configuring Segment Routing Traffic Engineering With OSPF 96
Configuring Intra Area Tunnel 96
Configuring Inter Area Tunnel 99
Verifying Configuration of the SR-TE Tunnels 102
Verifying Tunnel 1 102
Verifying Tunnel 2 102
Verifying Tunnel 3 103
Verifying Tunnel 4 104
Verifying Tunnel 5 104

CHAPTER 9 **BGP Dynamic Segment Routing Traffic Engineering** 107
Feature Information for BGP Dynamic Segment Routing Traffic Engineering 107

Restrictions for Segment Routing –Traffic-Engineering Dynamic BGP 107
Information About Segment Routing –Traffic-Engineering Dynamic BGP 108
TE Label Switched Path Attribute-Set 109
How to Configure TE Label Switched Path Attribute-Set 109
Configuring TE Label Switched Path Attribute-Set 109

CHAPTER 10 **Segment Routing On Demand Next Hop for L3/L3VPN 111**
Feature Information for Segment Routing On Demand Next Hop for L3/L3VPN 111
Restrictions for Segment Routing On Demand SR PFP ODN AUTO STEERING (PCE DELEGATED) for L3/L3VPN 112
Information About Segment Routing On Demand SR PFP ODN AUTO STEERING (PCE DELEGATED) for L3/L3VPN 112
How to Configure Segment Routing On Demand Next Hop for L3/L3VPN 113
Configuring Segment Routing On Demand Next Hop for L3/L3VPN 113
Verifying Segment Routing On Demand Next Hop for L3/L3VPN 117

CHAPTER 11 **Segment Routing On Demand for L2VPN/VPWS 123**
Feature Information for Segment Routing On Demand Next Hop for L2VPN/VPWS 123
Restrictions for Segment Routing On Demand Next Hop for L2VPN/VPWS 124
Information About Segment Routing On Demand Next Hop for L2VPN/VPWS 124
AToM Manager 125
Inter-Area L2VPN ODN 125
How to Configure Segment Routing On Demand Next Hop for L2VPN/VPWS 125
Configuring Segment Routing On Demand Next Hop for L2VPN/VPWS Using Pesudowire Interface Commands 125
Configuring Segment Routing On Demand Next Hop for L2VPN/VPWS Using Template Commands 126
Configuring Segment Routing On Demand Next Hop for L2VPN/VPWS With Prepend Option 127
Configuring Preferred Path for Segment Routing On Demand Next Hop for L2VPN/VPWS 127
Configuring Autoroute Destination for Segment Routing On Demand Next Hop for L2VPN/VPWS 128
Verifying Segment Routing On Demand Next Hop for L2VPN/VPWS 128

CHAPTER 12 **Fast Convergence Default Optimize 133**
Feature Information for Fast Convergence Default Optimize 133

Information About Fast Convergence Default Optimize 133
Default Optimize Values for IS-IS 134
Default Optimize Values for OSPF 135

CHAPTER 13 Routing Information Base Support 137
Feature Information for Routing Information Base Support 137
Routing Information Base Support for Route Redistribution 138
OSPF Node SID Redistribution Support 138
Information About OSPF Node SID Redistribution Support 138
NSSA ASBR 138
non-NSSA ASBR 138
Redistributing Prefix 139
Verify OSPF Node SID Redistribution 139
Routing Information Base Support for On-Demand Next Hop 140

CHAPTER 14 SR-TE On Demand LSP 143
Feature Information for SR-TE On Demand LSP 143
Restrictions for SR-TE On Demand LSP 144
Information About SR-TE On Demand LSP 144
SR-TE: Setup LSP as Static Route 144
Static SRTE over Unnumbered Interfaces 145
How to Configure SR-TE On Demand LSP 145
Configuring LSP as Static Route 145
Enabling Segment Routing Auto Tunnel Static Route 145
Verifying Segment Routing Auto-Tunnel Static Route 146
Configure Native UCMP for Static Routing 148
Local UCMP 148
Native UCMP 148
Configuration Example 149

CHAPTER 15 Segment Routing MPLS OAM Support 151
Feature Information for Segment Routing OAM Support 151
Restrictions for Segment Routing OAM MPLS Support 152
Information About Segment Routing MPLS OAM Support 152

Segment Routing OAM Support 152
Benefits of Segment Routing OAM Support 152
Segment Routing MPLS Ping 153
Segment Routing MPLS Traceroute 153
LSP Ping Operation for Nil FEC target 153
How to Diagnose Segment Routing with LSP Ping and Trace Route Nil FEC Target 154
Using LSP Ping for Nil FEC Target 154
Using LSP Traceroute for Nil FEC Target 154
Example for LSP Ping Nil FEC Target Support 154
Path Validation in Segment Routing Network 156
MPLS Ping and Traceroute for IGP Prefix-SID FEC Type 156
MPLS Ping and Traceroute for IGP-Adjacency Segment ID 158
Configuring Segment Routing MPLS Traffic Engineering for MPLS Ping and Traceroute 158
Configuring Segment Routing MPLS IGP for MPLS Ping and Traceroute 159

CHAPTER 16 **Using Seamless BFD with Segment Routing 161**
Feature Information for Seamless BFD with Segment Routing 161
Restrictions For Using Seamless BFD with Segment Routing 162
Information About Seamless BFD with Segment Routing 162
Bidirectional Forwarding Detection and Seamless-Bidirectional Forwarding Detection (S-BFD) 162
Initiators and Reflectors 162
How to Configure Seamless BFD with Segment Routing 163
Configuring Seamless-Bidirectional Forwarding Detection (S-BFD) for Segment Routing 163
Enabling Seamless Bidirectional Forwarding Detection (S-BFD) on the Reflector Node 164
Enabling Seamless Bidirectional Forwarding Detection (S-BFD) on the Initiator Node 164
Enabling Segment Routing Traffic Engineering Tunnel with Seamless-Bidirectional Forwarding (S-BFD) 164
Verifying S-BFD Configuration 164
Additional References for Seamless BFD with Segment Routing 165

CHAPTER 17 **Using SSPF with Segment Routing 167**
Feature Information for SSPF with Segment Routing 167
Information About SSPF with Segment Routing 167
Strict Shortest Path First 167

Approaches for Configure Strict Shortest Path First 168
How to Configure SSPF with Segment Routing 168
Configuring Strict Shortest Path First (SPF) 168
Enabling Strict Shortest Path First Using the connect-prefix-sid-map command 168
Enabling Strict Shortest Path First Using Segment Routing Mapping Server 169
Additional References for SSPF with Segment Routing 170

CHAPTER 18 **Dynamic PCC** 171
Information About Dynamic PCC 171
Path Computation Element Protocol Functions 171
Redundant Path Computation Elements 171
How to Configure Dynamic PCC 172
Configuring Dynamic PCC Globally 172
Configuring Dynamic PCC on an Interface 172
Configuring Dynamic PCC With Verbatim Path Option 172
Verifying Dynamic PCC 173
Verifying Verbatim Path Option With Dynamic PCC 176
Feature Information for Dynamic PCC 177

CHAPTER 19 **SR: PCE Initiated LSPs** 179
Prerequisites for SR: PCE Initiated LSPs 179
Restrictions for SR: PCE Initiated LSPs 179
Information About SR: PCE Initiated LSPs 179
Overview of Path Computation Element Protocol 179
SR: PCE Initiated LSPs 180
Single and Redundant PCE Operations 180
How to Configure SR: PCE Initiated LSPs 181
Establishing a PCEP session with PCC 181
Advertising an LSP in a Network 181
Specifying Precedence of a PCE for PCC 181
Verifying LSP Configurations 182
Additional References for SR: PCE Initiated LSPs 187
Feature Information for SR: PCE Initiated LSPs 187

CHAPTER 20 **ISIS - SR: uLoop Avoidance** 189

Prerequisites for ISIS - SR: uLoop Avoidance 189

Restrictions for ISIS - SR: uLoop Avoidance 189

Information About ISIS - SR: uLoop Avoidance 189

Microloops 189

Segment Routing and Microloops 192

How Segment Routing Prevents Microloops? 192

How to Enable ISIS - SR: uLoop Avoidance 193

Enabling Microloop Avoidance 193

Verifying Microloop Avoidance 193

Additional References for ISIS - SR: uLoop Avoidance 194

Feature Information for ISIS - SR: uLoop Avoidance 194

CHAPTER 21 **BGP - SR: BGP Prefix SID Redistribution** 197

Prerequisites for BGP - SR: BGP Prefix SID Redistribution 197

Information About BGP - SR: BGP Prefix SID Redistribution 197

Segment Routing and BGP 197

Segment Routing for Locally Sourced Routes 198

Segment Routing for Received Prefixes 198

Segment Routing for Redistributed Routes 198

BGP--MFI Interaction 198

How to Enable BGP - SR: BGP Prefix SID Redistribution 198

Enabling BGP-Prefix-SID 198

Enabling BGP for Segment Routing 199

Verifying BGP - SR: BGP Prefix SID Redistribution 199

Additional References for BGP - SR: BGP Prefix SID Redistribution 200

Feature Information for BGP - SR: BGP Prefix SID Redistribution 200

CHAPTER 22 **Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS** 201

Restrictions for Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS 201

Information About Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS 201

Maximum SID Depth 201

Node Maximum SID Depth Advertisement 202

Getting the Node MSD from Hardware 203
Advertising the MSD to BGP-LS 203
Verifying Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS 203
Feature Information for Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS 204

CHAPTER 23 RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel 205
Feature Information for RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel 205
Prerequisites for RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel 206
Restrictions for RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel 206
Information About RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel 207
Benefits of RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel 207
Backup AutoTunnel 207
How to Configure RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel 209
Configuring Explicit Path for Point-to-Point Network Type 209
Configuring Explicit RSVP-TE Tunnel With FRR 210
Verifying RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel 211

CHAPTER 24 ISIS Manual Adjacency SID 213
Feature Information for ISIS Manual Adjacency SID 213
Information About ISIS Manual Adjacency SID 213
Manual Adjacency SID 214
Adjacency SID Advertisement 214
Adjacency SID Forwarding 215
Configuration Prerequisites 215
Configuring Manual Adjacency SID 215
Verifying Manual Adjacency SID 216

CHAPTER 25 OSPF Manual Adjacency SID 217
Feature Information for OSPF Manual Adjacency SID 217
Information About OSPF Manual Adjacency SID 217
Prerequisites for OSPF Manual Adjacency SID 218
Restrictions for OSPF Manual Adjacency SID 218
Manual Adjacency SIDs 218

Manual Adjacency SID Advertisement 219
Manual Adjacency SID Forwarding 219
How to Configure OSPF Manual Adjacency SID 219
Modifying Segment Routing Local Block Range 219
Configuring OSPF Manual Adjacency SID 219
Verifying OSPF Manual Adjacency SID 220

CHAPTER 26 **OSPFv2 Segment Routing Strict SPF 223**
Feature Information for OSPFv2 Segment Routing Strict SPF 223
Restrictions for OSPFv2 Segment Routing Strict SPF 224
Information About OSPFv2 Segment Routing Strict SPF 224
Why Strict SPF 224
Strict-SPF Capability Advertisement 224
Strict-SPF SID Advertisement in Extended Prefix LSA 225
Interaction with SR-TE and Router Information Base 225
Enabling and Disabling OSPFv2 Segment Routing Strict SPF 225
Configuring OSPFv2 Segment Routing Strict SPF SID 226
Verifying OSPFv2 Segment Routing Strict SPF 226

CHAPTER 27 **Segment Routing OSPFv2 Microloop Avoidance 233**
Feature Information for Segment Routing OSPFv2 Microloop Avoidance 233
Information About Segment Routing OSPFv2 Microloop Avoidance 234
Microloops 234
Preventing Microloops using Segment Routing 237
Prerequisites for Segment Routing OSPFv2 Microloop Avoidance 237
Restrictions for Segment Routing OSPFv2 Microloop Avoidance 238
Configuring Segment Routing OSPFv2 Microloop Avoidance 238
Verifying Segment Routing OSPFv2 Microloop Avoidance 238

CHAPTER 28 **Performance Measurement for Traffic Engineering 239**
Feature Information for Performance Measurement for Traffic Engineering 239
Information about Performance Metrics for Traffic Engineering 240
Overview of Link Delay Measurement 240
Link Delay Metrics for a Computation Interval 240

Link Delay Metrics for Advertisement 241
Global Link Delay Profile 242
Benefits of Link Delay Measurement 243
Restrictions for Link Delay Measurement 243
How to Configure Performance Measurement for Traffic Engineering 244
Configuring Global Link Delay Profile 244
Configuring Link Delay Measurement for an Interface 244
Enabling Monitoring Mode 245
Verifying Link Delay Configuration 246
Viewing Link Delay Information for an Interface 246
Additional Commands 247
Additional References 249

CHAPTER 29 **Configure Performance Measurement** 251
Link Delay Measurement 252
Restrictions and Usage Guidelines for PM for Link Delay 253
PM Link Delay: Default Values for Different Parameters 253
Configuration Example: PM for Link Delay 253
Verification: PM Link Delay Configuration 254
End-to-End Delay Measurement 257
Configuration Example: PM for End-to-End Delay Management 258
Verification: PM End-to-End Delay Management Configuration 259
One-Way Link Loss Measurement 260
Information About One-Way Link Loss Measurement 260
Restrictions for One-Way Link Loss Measurement 261
Supported Platforms for One-Way Link Loss Measurement 261
Dual-Color Loss Measurement for GRE-IPSec Tunnel 262
IGP IS-IS Advertisement for Link Loss Measurement 263
Configuration Example: One-Way Link Loss Measurement 263
Configuration Example: SR-MPLS Policy Configuration 265
Verification: One-Way Link Loss Measurement 265
Debugging and Troubleshooting One-Way Link Loss Measurement 269
Sample show Commands 270

CHAPTER 30 **IP Endpoint Delay Measurement and Liveness Monitoring** 275

Information About IP Endpoint Performance Delay Measurement and Liveness Monitoring 275

Benefits of IP Endpoint Performance Delay Measurement and Liveness Monitoring 275

Restrictions for IP Endpoint Performance Delay Measurement and Liveness Monitoring 276

Supported Platforms for IP Endpoint Performance Delay Measurement and Liveness Monitoring 276

Use Cases for IP Endpoint Performance Delay Measurement 276

Use Case 1: Delay Measurement Probe Toward an IP Endpoint Reachable in the Global Routing Table 277

Use Case 2: Delay Measurement Probe Toward an IP Endpoint Reachable in a User-Specified VRF 277

How to Configure IP Endpoint Performance Delay Measurement 278

Usage Guidelines and Limitations 278

Configuring Performance Delay Measurement IP Endpoint 279

Configuring IP Endpoint Performance Delay Measurement Profile 280

Configuration Examples for IP Endpoint Performance Delay Measurement 283

Configuration Example: Configuring IP Endpoint Performance Delay Measurement (global configuration) 284

Configuration Example: Configuring IP Endpoint Performance Delay Measurement (VRF configuration) 285

Verification for IP Endpoint Performance Delay Measurement 285

Examples 286

show performance-measurement summary 286

show performance-measurement profile 286

show performance-measurement endpoint session 287

show performance-measurement counters endpoint 288

show performance-measurement history endpoint 288

Feature Information for IP Endpoint Delay Measurement and Liveness Monitoring 289

CHAPTER 31 **SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated)** 291

Feature Information for SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated) 291

Information About SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated) 292

Restrictions for SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated) 293

BGP Color Extended Community and VRF Prefix Coloring 293

Supported Platforms 294
Attaching a Color-Extended Community 294
Support for PFP with RIB Path 296
Example: Configuring PFP with RIB Path 296
Configuring SR-TE Per-Flow Class (ODN) and Automated Steering (PCE Delegated) 297
Verifying SR-TE Per-Flow Class (ODN) and Automated Steering (PCE Delegated) 299

CHAPTER 32 **Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains** 301
Feature Information for Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains 301
Information about Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains 302
Overview of the Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains 302
How to Configure Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains 302
Configure the Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains 302
Verify the Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains 303
Example: Configure Loopback Prefix SIDs of a BR in Multiple ISIS Domains 304

CHAPTER 33 **Traffic Steering by Dropping Invalid Paths** 307
Feature Information for Traffic Steering by Dropping Invalid Paths 307
Overview 308
Before You Begin 308
Benefits 308
Restrictions 308
How to Configure Traffic Steering by Dropping Invalid Paths 308
Configuring for a PCC Profile 308
Configuring for Static Policies 309
Configuring for On-Demand Next Hop for SR-TE Policies 309
Show Commands 309

CHAPTER 34 **Configuring the Cisco ISIS Local Unequal Cost Multipath (UCMP)** 311
Configuring the Unequal Cost Multi Path (UCMP) Local 312

Verifying the Unequal Cost Multi Path (UCMP) Local 312
Examples: Show Commands 312
Debug Commands 313
Feature Information for Segment Routing—IS-IS UCMP 313

CHAPTER 35 **Enabling Segment Routing Flexible Algorithm 315**
Feature History 316
Prerequisites for Flexible Algorithm 317
Restrictions for Flexible Algorithm 317
Building Blocks of Segment Routing Flexible Algorithm 317
Flexible Algorithm Definition 317
Flexible Algorithm Support Advertisement 318
Flexible Algorithm Definition Advertisement 318
Flexible Algorithm Prefix-SID Advertisement 318
Calculation of Flexible Algorithm Path 319
Installation of Forwarding Entries for Flexible Algorithm Paths 319
Flexible Algorithm Prefix-SID Redistribution 319
Displaying the Algorithm Information 320
Flexible Algorithm Prefix Metric Advertisement 320
Flexible Algorithm Configurations 321
Configuring IS-IS Flexible Algorithm 323
Redistributing IS-IS 325
Configuring SRTE-ODN Association 325
Configuring the Interface for Flexible Algorithm 325
Configuring BGP 325
Configuring Selective Path Filtering 326
Configuring SR Policy with PCE Delegation 326
Verifying the Flexible Algorithm Configuration 327

CHAPTER 36 **L2VPN over SR-TE Preferred Path 333**
Restrictions 334
Configuring L2VPN Traffic Steering Using SR-TE Preferred Path with Flexible Algorithm 334
Configuration Example 1: VPWS Psuedowire over SR-TE Preferred Path 336
Configuration Example 2: VPWS Psuedowire over SR-TE Preferred Path 336

Configuration Example 3: VPLS Psuedowire over SR-TE Preferred Path 337
Verification of L2VPN over SR-TE Preferred Path Configuration 337

CHAPTER 37 COE-PCE Initiated SR Policy with IGP Autoroute Announce 339
COE-PCE Initiated SR Policy 340
Restrictions for PCE Initiated SR Policy 340
ECMP Over SR-TE 341
Restrictions for ECMP over SR-TE Policies 341
Local Congestion Mitigation 342
Load Balancing 343
Autoroute Announcement 343
Static Route Configuration 344
Next Hop ECMP within a SR Policy 344
Configuring with IGP Autoroute Announce 344
Verifying SR Policy with Autoroute Announce 344
Verifying ISIS Autoroute for IGP 345
Verify the Tunnel ID on the SR Policy 345

CHAPTER 38 DC-PE Router in Cisco ACI to SR-MPLS Hand-off 347
Prerequisites 347
Restrictions 347
Information About DC-PE Router in ACI to SR-MPLS Hand-off 347
Supported Platforms 348
How to Configure the DC-PE Router 348
Configuring VRF on the DC-PE Router 349
Configuring BGP on the DC-PE router. 351
Verifying DC-PE Router Configuration 356
Verifying IPv4 and IPv6 Route from ACI 356
Verifying IPv4 and IPv6 Route from WAN 358
Troubleshooting and Debugging 359
Feature Information for DC-PE Router in Cisco ACI to SR-MPLS Hand-off 360

CHAPTER 39 Segment Routing over IPv6 361
Segment Routing over IPv6 362

Feature Information 362
Restrictions for SRv6 363
Information About SRv6 363
SRv6 Micro-Segment (uSID) 364
SRv6 Implementation 365
Supported Platforms 366
Configuring SRv6 366
Configuring SRv6 366
Verifying SRv6 Configuration 367
SRv6 under IS-IS 370
SRv6 under IS-IS 370
Information About SRv6 under IS-IS 371
Configuring SRv6 under IS-IS 371
Verifying SRv6 IS-IS Configuration 371
SRv6 BGP-Based Services 372
SRv6 BGP-Based Services 372
Restrictions for SRv6 BGP-Based Services 373
Information About SRv6 BGP-Based Services 373
SRv6 Based L3VPN 374
Configuring SRv6 based L3VPN 374
SRv6 eBGP Inter-AS 376
BGP MPLS and SRv6 Co-Existence 378
Configuring MPLS and SRv6 Coexistence for L3VPN 379
Verifying SRv6 State 379
Troubleshooting and Debugging SRv6 BGP 384
EVPN VPWS over SRv6 385
Feature history for EVPN VPWS over SRv6 385
EVPN VPWS over SRv6 385
Configure EVPN VPWS over SRv6 386
EVPN VPWS over SRv6 configuration examples 388
Verify EVPN VPWS over SRv6 391
BGP SRv6 L3VPN On-Demand Next-Hop 396
BGP SRv6L3VPN On-Demand Next-Hop 396
Prerequisites for BGP SRv6 L3VPN ODN 396

Information About BGP SRv6L3VPN ODN 396
Configuring SRv6 L3VPN ODN 397
Configuring SRv6 ODN Color Template 398
Verifying SRv6 L3VPN ODN Configuration 401
Debugging SRv6 L3VPN ODN Configuration 404
SRv6 Traffic Engineering Policies 406
SRv6 Traffic Engineering Policies 406
SRv6 Traffic Engineering Policies 406
Restrictions for SRv6-TE Policies 406
Information About SRv6-TE Policies 407
Configuring SRv6-TE 407
Verifying SRv6-TE Configuration 409
Troubleshooting and Debugging SRv6-TE 412
Performance Measurement for SRv6 413
Performance Measurement for SRv6 413
Performance Measurement for SRv6 413
Performance Measurement Liveness for SRv6 413
Configuring PM Liveness for SRv6 413
Verifying Performance Measurement for SRv6 414
SRv6 OAM 419
SRv6 Operations, Administration, and Maintenance 419
Restrictions for SRv6 419
Information About SRv6 OAM 419
Operating SRv6 OAM 420
Support for SRv6 OAM-TE 424
Troubleshooting and Debugging SRv6 OAM-TE 425

CHAPTER 40 **ISIS - SRv6: uLoop Avoidance** 427
Prerequisites for ISIS - SRv6: uLoop Avoidance 427
Restrictions for ISIS - SRv6: uLoop Avoidance 427
Information About ISIS - SRv6: uLoop Avoidance 427
Microloops 427
SRv6 and Microloops 430
How Segment Routing Prevents Microloops 431
Supported Platforms 432
How to Enable ISIS - SRv6: uLoop Avoidance 432

Configuring uLoop Avoidance 432
Verifying Microloop Avoidance 432
Additional References for ISIS - SRv6: uLoop Avoidance 433
Feature Information for ISIS - SRv6: uLoop Avoidance 434

CHAPTER 41 **IPv6 Loop-Free Alternate Fast Reroute 435**
Prerequisites for IPv6 LFA FRR 435
Restrictions for IPv6 LFA FRR 435
Information About IPv6 LFA FRR 436
IS-IS and IPv6 FRR 436
Repair Paths 436
LFA Overview 437
LFA Calculation 437
Interaction Between RIB and Routing Protocols 437
How to Configure IPv6 LFA FRR 438
Configuring FRR Support 438
Additional IS-IS IPv6 Commands 440
Configuration Examples for IPv6 LFA FRR 441
Example: Configuring IPv6 LFA FRR 441
Verifying IPv6 LFA FRR Configuration 442
Feature Information for Configuring IPv6 LFA FRR 442

CHAPTER 42 **IS-IS SRv6 Link-protection Topology Independent Loop Free Alternate Fast Reroute 445**
Feature Information for IS-IS SRv6 Link-protection TI-LFA FRR 445
Prerequisites for IS-IS SRv6 Link-protection TI-LFA FRR 445
Restrictions for IS-IS SRv6 Link-protection TI-LFA FRR 446
Information About IS-IS SRv6 Link-protection TI-LFA FRR 446
Topology-Independent Loop Free Alternate 446
Topology Independent Loop Free Alternate Tie-break 447
Interface Fast Reroute Tiebreakers 447
How to Configure IS-IS SRv6 Link-protection TI-LFA FRR 448
Configuring Topology Independent Loop Free Alternate Fast Reroute 448
Examples: Configuring IS-IS SRv6 Link-protection TI-LFA FRR 449
Verifying the Tiebreaker 450

Verifying the Primary and Repair Paths 450
Verifying SRv6 Configuration 452
Verifying the IS-IS Topology Independent Loop Free Alternate Paths 452

CHAPTER 43 SRv6 Flex-Algo with TI-LFA and uLoop Avoidance 455
Feature Information for SRv6 Flex-Algo with TI-LFA and uLoop Avoidance 455
SRv6 Flex-Algo with TI-LFA and uLoop Avoidance 455
SRv6 flex-algo FRR and TI-LFA 456
SRv6 Flex-Algo Microloop-Avoidance 456
Prerequisites for SRv6 Flex-Algo with TI-LFA and uLoop Avoidance 456
Restrictions for SRv6 Flex-Algo with TI-LFA and uLoop Avoidance 456
Configure SRv6 IS-IS flexible algorithm 457
Verify SRv6 Flex-Algo with TI-LFA and uLoop Avoidance 458
Monitor SRv6 Flex-Algo with TI-LFA and uLoop Avoidance 460
Traffic steering 461
Configure traffic steering 461

CHAPTER 44 SRv6 Path MTU Discovery on IOS XE devices 463
Feature Information SRv6 Path MTU Discovery 463
SRv6 Path MTU Discovery 464
Relaying ICMP PTB messages 464
Transit-node PTB relay 464
Headend-node PTB relay 465
Restrictions for SRv6 path MTU discovery 466
Configure SRv6 encapsulation MTU 466
Enable Transit-Node PTB relay 466
Verify SRv6 path MTU discovery 467
Supported SRv6 PMTU Commands 468

CHAPTER 45 MVPN ingress replication over SRv6 471
Feature Information for MVPN ingress replication over SRv6 471
MVPN Ingress Replication over SRv6 471
How MVPN Ingress Replication over SRv6 works 472
Partitioned MDT profiles 473

Configure MVPN ingress replication over SRv6 **473**

Configure SRv6 MVPN multicast **474**

Configure BGP MVPNv4 neighbor encapsulation **475**

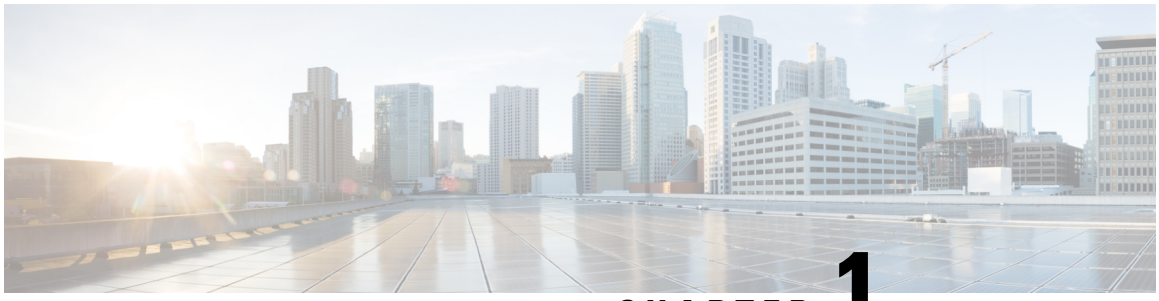
BGP SRv6 configuration **475**

Verify and monitor MVPN ingress replication over SRv6 **476**

Appendix: configuration reference **487**

SRv6 configuration for PE11 **487**

SRv6 configuration for PE12 **498**



CHAPTER 1

Read Me First

Important Information

Note For CUBE feature support information in Cisco IOS XE Bengaluru 17.6.1a and later releases, see Cisco Unified Border Element IOS-XE Configuration Guide.

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Feature Information

Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

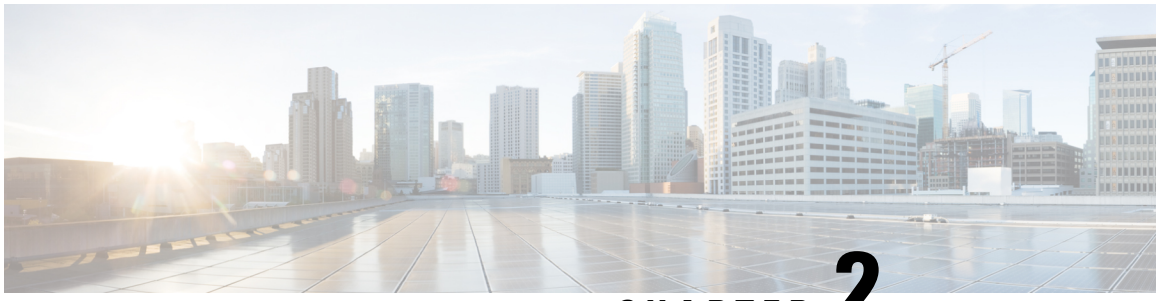
Obtaining Documentation and Submitting a Service Request

- 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.

• Short Description, on page 2

Short Description

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)



CHAPTER 2

Introduction to Segment Routing for the MPLS Dataplane

This chapter introduces the concept of Segment Routing and contains the following sections:

- Feature Information for Segment Routing, on page 3
- Overview of Segment Routing for the MPLS Dataplane, on page 3
- How Segment Routing Works, on page 4
- Examples for Segment Routing, on page 5
- Benefits of Segment Routing, on page 6
- Segment Routing Global Block, on page 8
- Additional References for Segment Routing, on page 9

Feature Information for Segment Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 1: Feature Information for Segment Routing

Feature Name	Releases	Feature Information
Introduction to Segment Routing	Cisco IOS XE Amsterdam 17.3.2	Segment Routing is a flexible, scalable way of doing source routing.

Overview of Segment Routing for the MPLS Dataplane

Segment Routing (SR) is a flexible, scalable way of doing source routing. The source chooses a path and encodes it in the packet header as an ordered list of segments. Segments are an identifier for any type of instruction. Each segment is identified by a segment ID (SID) . A segment instruction can be:

- Go to node N using the shortest path.

- Go to node N over the shortest path to node M and then follow a specific set of links.
- Apply service S.

With segment routing, the network no longer needs to maintain a per-application and per-flow state. Instead, it obeys the forwarding instructions provided in the packet.

Segment Routing relies on extensions to the Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) protocols. When operating using the MPLS (Multiprotocol Label Switching) dataplane it integrates with the rich multi service capabilities of MPLS, including Layer 3 VPN (L3VPN), Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS), and Ethernet VPN (EVPN).



Note The Cisco IOS XE Amsterdam 17.3.2 release only provides support for the MPLS dataplane for the IPv4 address family.

Segment routing can be directly applied to the Multiprotocol Label Switching (MPLS) architecture with no change in the forwarding plane. Segment routing utilizes the network bandwidth more effectively than traditional MPLS networks and offers lower latency. A segment is encoded as an MPLS label. An ordered list of segments is encoded as a stack of labels. The segment to process is on the top of the stack. The related label is popped from the stack, after the completion of a segment.

Segment Routing provides automatic traffic protection without any topological restrictions. The network protectstraffic against link and node failures without requiring additional signaling in the network. Existing IP fast re-route (FRR) technology, in combination with the explicit routing capabilities in Segment Routing guarantees full protection coverage with optimum backup paths. Traffic protection does not impose any additional signaling requirements.

How Segment Routing Works

A router in a Segment Routing network is capable of selecting any path to forward traffic, whether it is an explicit path, or an Interior Gateway Protocol (IGP) calculated shortest path. Segments represent subpaths that a router can combine to form a complete path to a network destination. Each segment has an identifier (Segment Identifier) that is advertised throughout the network using IGP extensions. Unlike the case for traditional MPLS networks, routers in a Segment Router network do not require Label Distribution Protocol (LDP) and Resource Reservation Protocol - Traffic Engineering (RSVP-TE) to support forwarding.

Each router (node) and each link (adjacency) have an associated segment identifier (SID). Node segment identifiers are globally unique. The network administrator allocates a node ID to each router from a reserved block. On the other hand, an adjacency segment ID is locally significant to the advertising node and represents a specific adjacency, such as an egress interface, to a neighboring router. Routers automatically generate adjacency identifiers outside of the reserved block used for node IDs. In an MPLS network, a segment identifier is encoded as an MPLS label stack entry. Segment IDs direct data packets along a specified path. There are two kinds of segment IDS:

- **Prefix SID**:

  A segment ID that is associated with an IP address prefix associated with a node in the network. Prefix SIDs are globally unique. A node segment is a special prefix segment that is bound to the loopback address of a node. It is advertised as an index into node specific SR Global Blocks (SRGBs).

- **Adjacency SID**:

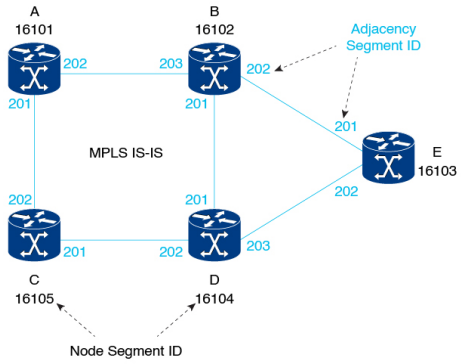
A segment ID that is associated with a link between two routers. Adjacency SIDs are scoped to a specific router.

A node segment can be a multihop path while an adjacency segment is always a one-hop path.

Examples for Segment Routing

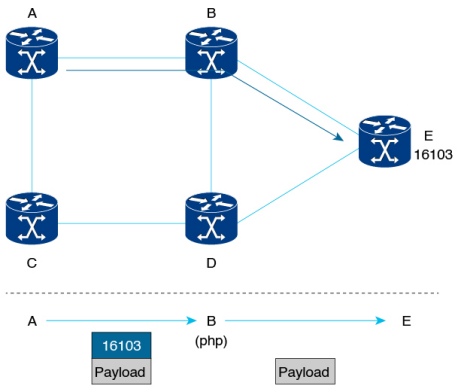
The following figure illustrates an MPLS network with five routers using Segment Routing, IS-IS, a label range of 16000 to 23999 for node IDs, and 200 and higher for adjacency IDs. IS-IS would distribute IP prefix reachability alongside segment ID (the MPLS label) across the network.

Figure 1: An MPLS Network with Five Routers Using Segment Routing



In Figure 1, any router sending traffic to router E would push label 16103 (router E node segment identifier) to forward traffic using the IS-IS shortest path. The MPLS label-swapping operation at each hop preserves label 16103 until the packet arrives at E (Figure 2). On the other hand, adjacency segments behave differently. For example, if a packet arrives at Router D with a top-of-stack MPLS label of 203 (D-to-E adjacency segment identifier), Router D would pop the label and forward the traffic to Router E.

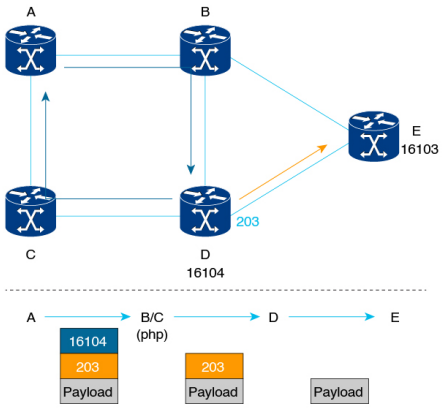
Figure 2: MPLS Label-Swapping Operation



Segment identifiers can be combined as an ordered list to perform traffic engineering. A segment list can contain several adjacency segments, several node segments, or a combination of both depending on the forwarding requirements. In the previous example, Router A could alternatively push label stack (16104, 203) to reach Router E using the shortest path and all applicable ECMPs to Router D and then through an explicit interface onto the destination (Figure 3). Router A does not need to signal the new path, and the state information

remains constant in the network. Router A ultimately enforces a forwarding policy that determines which flows destined to router E are switched through a particular path.

Figure 3: Router E Destination Path



Benefits of Segment Routing

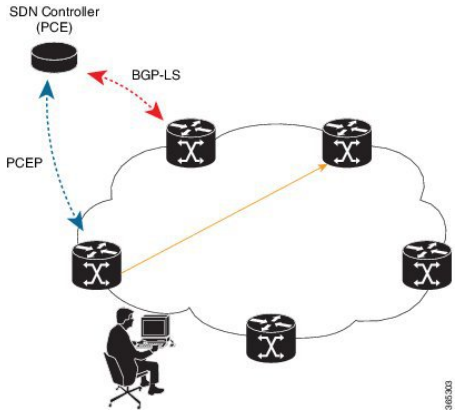
- **Ready for SDN**: Segment Routing is a compelling architecture conceived to embrace Software-Defined Network (SDN) and is the foundation for Application Engineered Routing (AER). It strikes a balance between network-based distributed intelligence, such as automatic link and node protection, and controller-based centralized intelligence, such as traffic optimization.

  It can provide strict network performance guarantees, efficient use of network resources, and very high scalability for application-based transactions. The network uses minimal state information to meet these requirements. Segment routing can be easily integrated with a controller-based SDN architecture.

  The following figure illustrates a sample SDN scenario where the controller performs centralized optimization, including bandwidth admission control. In this scenario, the controller has a complete picture of the network topology and flows. A router can request a path to a destination with certain characteristics, for example, delay, bandwidth, diversity. The controller computes an optimal path and returns the corresponding segment list, such as an MPLS label stack, to the requesting router. At that point, the router can inject traffic with the segment list without any additional signaling in the network.

  In addition, segment lists allow complete network virtualization without adding any application state to the network. The state is encoded in the packet as a list of segments. Because the network only maintains segment state, it can support a large number - and a higher frequency - of transaction-based application requests without creating any burden on the network.

Figure 4: SDN Controller



- **Simplified Operation**:
  - When applied to the MPLS data plane, Segment Routing offers the ability to tunnel MPLS services (VPN, VPLS, and VPWS) from an ingress provider edge to an egress provider edge without any other protocol than an IGP (ISIS or OSPF).
  - Simpler operation without separate protocols for label distribution (for example, no LDP or RSVP).
  - No complex LDP or IGP synchronization to troubleshoot.
  - Better utilization of installed infrastructure, for lower capital expenditures (CapEx), with ECMP-aware shortest path forwarding (using node segment IDs).

- **Supports Fast Reroute (FRR)**: Deliver automated FRR for any topology. In case of link or node failures in a network, MPLS uses the FRR mechanism for convergence. With segment routing, the convergence time is sub-50-msec.

- **Large-Scale Data Center**:
  - Segment Routing simplifies MPLS-enabled data center designs using Border Gateway Protocol (BGP) RFC 3107 - IPv4 labeled unicast among Top-of-the-Rack/Leaf/Spine switches.
  - BGP distributes the node segment ID, equivalent to IGP node SID.
  - Any node within the topology allocates the same BGP segment for the same switch.
  - The same benefits are provided as for IGP node SID: ECMP and automated FRR (BGP PIC(Prefix Independent Convergence).
  - This is a building block for traffic engineering - SR TE data center fabric optimization.

- **Dual-Plane Networks**:
  - Segment Routing provides a simple solution for disjointness enforcement within a *dual-plane* network, where the route to an edge destination from a given plane stays within the plane unless the plane is partitioned.
  - An additional SID anycast segment ID allows the expression of macro policies such as: "Flow 1 injected in node A toward node Z must go via plane 1" and "Flow 2 injected in node A towards node Z must go via plane 2."

- **Centralized Traffic Engineering**:
  - Controllers and orchestration platforms can interact with segment routing traffic engineering for centralized optimization, such as WAN optimization.
  - Network changes such as congestion can trigger an application to optimize (recompute) the placement of segment routing traffic engineering tunnels.
  - Segment Routing tunnels are dynamically programmed onto the network from an orchestrator using southbound protocols like PCE.
  - Agile network programming is possible since Segment Routing tunnels do not require signaling and per-flow state at midpoints and tail end routers.
- **Egress Peering Traffic Engineering (EPE)**:
  - Segment Routing allows centralized EPE.
  - A controller instructs an ingress provider edge and content source to use a specific egress provider edge and specific external interface to reach a destination.
  - BGP peering segment IDs are used to express source-routed inter-domain paths.
  - Controllers learn BGP peering SIDs and the external topology of the egress border router through BGP Link Status (BGP-LS) EPE routes.
  - Controllers program ingress points with a desired path.
- **Plug-and-Play Deployment**: Segment routing tunnels are interoperable with existing MPLS control and data planes and can be implemented in an existing deployment.

Segment Routing Global Block

The Segment Routing Global Block (SRGB) is the range of labels reserved for segment routing globally scoped SIDs. SRGB is local property of a segment routing node. In the MPLS architecture, SRGB is the set of local labels reserved for global segments. In segment routing, each node can be configured with a different SRGB value and hence the absolute SID value associated to an IGP Prefix Segment can change from node to node.

The SRGB default range is 16000 to 23999. The SRGB can be configured as follows:

```
Device(config)# segment-routing mpls
Device(config-srmpls)#segment-routing global-block 45000 55000
```

The SRGB label value is calculated hop-by-hop as follows:

- Node-SIDs are advertised as an index into the local SRGB.
- IGPs calculate the appropriate MPLS label associated with a Node SID by adding the index to the SRGB advertised by the next hop.
- Platforms may have specific limitations on the lower and upper bounds for the SRGB. These bounds will be visible in the help string associated with the segment-routing global-block command e.g.:

```
Router(config-srmpls)#global-block ?
  <16-1048575>  SR GB/LB Label Range Start
```

Adjacency Segment Identifiers

An Adjacency Segment Identifier (adj-SID) is a local label that directs packets to a specific interface and a next hop. No specific configuration is required to enable adj-SIDs. Adjacency SIDs will be automatically allocated by the IGP for every neighbor.

Prefix Segment Identifiers

A prefix segment identifier (SID) is used to forward traffic to a destination represented by a prefix.

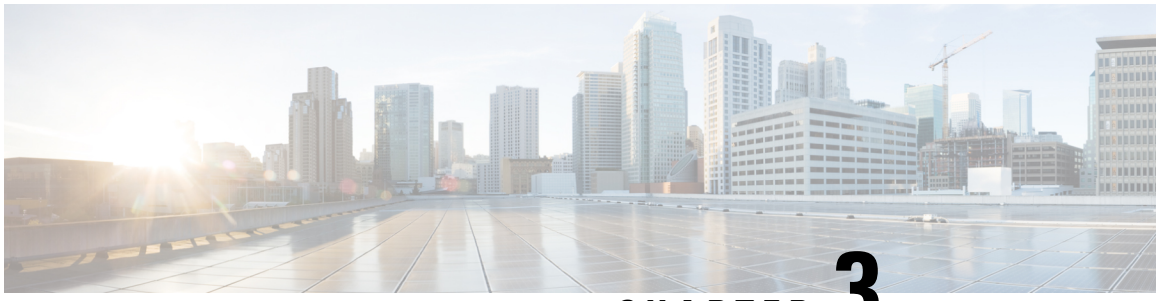
A prefix SID is an index into the Segment Routing Global Block (SRGB). The index maps to a local MPLS label, whose value is calculated for each node by adding the index value to the locally advertised SRGB starting value. For example:

- If a node advertises an SRGB range of 16000 – 23999, a SID index of 100 would map to MPLS label 16100.
- If a node advertises an SRGB range of 25000 - 39999, a SID index of 100 would map to MPLS label 25100

Additional References for Segment Routing

Related Documents

Related Topic	Document Title
Videos	• Introduction to Cisco Segment Routing (YouTube) • Introduction to Cisco Segment Routing (CCO)



CHAPTER 3

Segment Routing with IS-IS v4 Node SID

This chapter describes how Segment Routing works with IS-IS and contains the following sections:

- Feature Information for Segment Routing with IS-IS v4 Node SID, on page 11
- Restrictions for Segment Routing with IS-IS v4 Node SID, on page 11
- Information About Segment Routing with IS-IS v4 Node SID, on page 12
- How to Configure Segment Routing with IS-IS v4 Node SID, on page 15
- Configuration Examples for Segment Routing —IS-IS v4 Node SID, on page 23
- Additional References for Segment Routing with IS-IS v4 Node SID, on page 23

Feature Information for Segment Routing with IS-IS v4 Node SID

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Restrictions for Segment Routing with IS-IS v4 Node SID

- Segment routing must be configured at the top level before any routing protocol configuration is allowed under its router configuration sub mode.
- IS-IS protocol SR command is based on per topology (IPv4 address family).
- Effective Cisco IOS-XE Release 3.16, ISIS supports segment routing for IPv4 only.

Information About Segment Routing with IS-IS v4 Node SID

Segment Routing IS-IS v4 Node SID

Segment Routing relies on a small number of extensions to Cisco Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) protocols. There are two levels of configuration required to enable segment routing for a routing protocol instance. The top level segment routing configuration which is managed by segment routing infrastructure component enables segment routing, whereas, segment routing configuration at the router level enables segment routing for a specific address-family of a routing protocol instance. There are three segment routing states:

- SR_NOT_CONFIGURED
- SR_DISABLED
- SR_ENABLED

Segment routing configuration under the IGPs is allowed only if the SR state is either SR_DISABLED or SR_ENABLED. The SR_ENABLED state indicates that there is at least a valid SRGB range reserved through the MFI successfully. You can enable segment routing for IGPs under the router configuration sub mode, through commands. However, IGP segment routing are enabled only after the global SR is configured.

The SR_ENABLED is a necessary state for any protocol to enable SR, however, it is not a sufficient for enabling SR for a protocol instance. The reason being that the IS-IS still does not have any information about segment routing global block (SRGB) information. When the request to receive information about the SRGB is processed successfully, the IS-IS SR operational state is enabled.

Segment Routing requires each router to advertise its segment routing data-plane capability and the range of MPLS label values that are used for segment routing in the case where global SIDs are allocated. Data-plane capabilities and label ranges are advertised using the SR-capabilities sub-TLV inserted into the IS-IS Router Capability TLV-242 that is defined in RFC4971.

ISIS SR-capabilities sub TLV includes all reserved SRGB ranges. However, the Cisco implementation supports only one SRGB range. The supported IPv4 prefix-SID sub TLV are TLV-135 and TLV-235.

Prefix SIDs Received in Label-Switched Path from Remote Routers

Prefix SIDs received in a label switched path (LSP) with a reachability TLV (TLV 135 and 235) are downloaded to the routing information base (RIB) in the same way as BGP downloads per prefix VPN labels, only if the following conditions are met:

- Segment routing is enabled for the topology and address-family.
- Prefix-SID is valid.
- The local label binding to MFI is successful.

Note

- For SIDs that do not fit in the specified SID range, labels are not used when updating the RIB. For the cases, where SID fits in the SID range, but does not fit the next-hop neighbor SID range, remote label associated with that path is not installed.
- Node SIDs received in an LSP with reachability TLVs (TLV 135 and 235) are downloaded to RIB only if segment routing is enabled under the corresponding address-family.
- In case of multiple best next hops, if all the next hops do not support segment routing, ISIS treats the instance similar to mismatched labels assigned to the same prefix. That means, IS-IS ignores the labels and installs unlabeled paths for all ECMP paths into the global RIB.

Segment Routing Adjacency SID Advertisement

Effective with Cisco IOS-XE Release 3.17, IS-IS supports the advertisement of segment routing adjacency SID. An Adjacency Segment Identifier (Adj-SID) represents a router adjacency in Segment Routing.

A segment routing-capable router may allocate an Adj-SID for each of its adjacencies and an Adj-SID sub-TLV is defined to carry this SID in the Adjacency TLVs. IS-IS adjacencies are advertised using one of the IS-Neighbor TLVs below:

- TLV-22 [RFC5305]
- TLV-23 [RFC5311]

IS-IS allocates the adjacency SID for each IS-IS neighbor only if the IS-IS adjacency state is up and IS-IS segment routing internal operational state is enabled. If an adjacency SID allocation failure is due to out-of-label resource, IS-IS retries to allocate the Adj-SID periodically in a default interval (30 seconds).

Adjacency SIDs

Effective with Cisco IOS-XE Release 3.18, multiple adjacency-SIDs are supported. For each protected P2P/LAN adjacency, IS-IS allocates two Adj-SIDs. The backup Adj-SID is only allocated and advertised when FRR (local LFA) is enabled on the interface. If FRR is disabled, then the backup adjacency-SID is released. The persistence of protected adj-SID in forwarding plane is supported. When the primary link is down, IS-IS delays the release of its backup Adj-SID until the delay timer expires. This allows the forwarding plane to continue to forward the traffic through the backup path until the router is converged.

Cisco IOS-XE Release 3.18, IS-IS Adj-SID is changed to be per level based since the forwarding plane is unaware of protocol-specific levels. The allocated and advertised backup Adj-SIDs can be displayed in the output of **show isis neighbor detail** and **show isis data verbose** commands.

Segment Routing Mapping Server

Segment Routing Mapping Server (SRMS) allows configuration and maintenance of the Prefix-SID mapping policy entries. Effective with Cisco IOS-XE Release 3.17, the IGPs use the active policy of the SRMS to determine the SID values when programming the forwarding plane.

The SRMS provides prefixes to SID/Label mapping policy for the network. IGPs, on the other hand, are responsible for advertising prefixes to SID/Label mapping policy through the Prefix-SID/Label Binding TLV. Active policy information and changes are notified to the IGPs, which use active policy information to update forwarding information.

Connected Prefix SIDs

Sometimes, a router may install a prefix with a SID that is different than what it advertises to the LSP. For example, if more than one protocol or more than one IGP instance is announcing the same prefix with different SIDs to the SRMS, the SRMS resolves the conflict and announces the winning prefix and SID that may not be the same as the local instance. In that case, the IGP always advertises what it learns from its source LSP although it still tries to install the SID which may be different than what it learns in its LSP. This is done to prevent the IGP from redistributing the SIDs from another protocol or another protocol instance.

SRGB Range Changes

When IS-IS segment routing is configured, IS-IS must request an interaction with the SRGB before IS-IS SR operational state can be enabled. If no SRGB range is created, IS-IS will not be enabled.

When an SRGB change event occurs, IS-IS makes the corresponding changes in its sub-block entries. IS-IS also advertises the newly created or extended SRGB range in SR-capabilities sub-TLV and updates the prefix-sid sub TLV advertisement.



Note In Cisco IOS-XE Release 3.16 only one SRGB range and SRGB extension for the modification are supported.

SRGB Deletion

When IS-IS receives an SRGB deletion event, it looks for an SRGB entry in the IS-IS SRGB queue list. If an SRGB entry does not exist, IS-IS makes sure that there is no pending SRGB created event. If a pending SRGB creation event is found, then IS-IS removes the SRGB creation event, and completes the SRGB delete processing,

If an SRGB entry is found in the IS-IS SRGB queue, IS-IS locks the SRGB, redistributes the RIBs and un-advertises all prefixed-SIDs that have SID value within the pending delete SRGB range, and un-advertises the SRGB range from SR-capabilities sub TLV. Once IS-IS has completed the SRGB deletion processing, it unlocks the SRGB and deletes the SRGB from its SR sub-block entry.

If there is no valid SRGB after the deletion of the SRGB, IS-IS SR operational state becomes disabled.

MPLS Forwarding on an Interface

MPLS forwarding must be enabled before segment routing can use an interface. IS-IS is responsible for enabling MPLS forwarding on an interface.

When segment routing is enabled for a IS-IS topology, or IS-IS segment routing operational state is enabled, IS-IS enables MPLS for any interface on which the IS-IS topology is active. Similarly, when segment routing is disabled for a IS-IS topology, IS-IS disables the MPLS forwarding on all interfaces for that topology.

Segment Routing and LDP Preference

The command **sr-label-preferred** allows the forwarding interface to prefer the segment routing labels over LDP labels for all prefixes in a topology.

Segment Routing Traffic Engineering Announcements

IS-IS announces segment routing information to TE when it detects that both, IS-IS SR and TE are enabled for at least one level. IS-IS announce only the information that is obtained from the level for which TE is configured.

Similarly, IS-IS instructs TE to delete all announcements when it detects that segment routing is not enabled or TE is no longer configured on any level.

How to Configure Segment Routing with IS-IS v4 Node SID

Configuring Segment Routing

Before you begin

Before configuring IS-IS to support segment routing you must first configure the segment routing feature in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **segment-routing mpls**
4. **connected-prefix-sid-map**
5. **address-family ipv4**
6. **10.1.1.1/32 index 100 range 1**
7. **exit-address-family**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**segment-routing mpls** **Example:**	Enables the segment feature using the MPLS data plane.

	Command or Action	Purpose
	`Device(config-sr)# segment-routing mpls`	
Step 4	**connected-prefix-sid-map** **Example:** `Device(config-srmpls)# connected-prefix-sid-map`	Enters a sub-mode where you can configure address-family specific mappings for local prefixes and SIDs.
Step 5	**address-family ipv4** **Example:** `Device(config-srmpls-conn)# address-family ipv4`	Specifies IPv4 address prefixes.
Step 6	**10.1.1.1/32 index 100 range 1** **Example:** `Device(config-srmpls-conn-af)# 10.1.1.1/32 100 range 1`	Associates SID 100 with the address 10.1.1.1/32.
Step 7	**exit-address-family** **Example:** `Device(config-srmpls-conn-af)# exit-address-family`	Exits the address family.

Configuring Segment Routing on an IS-IS Network

Before you begin

Before you configure segment routing on IS-IS network, IS-IS must be enabled on your network.

SUMMARY STEPS

1. **router isis**
2. **net network-entity-title**
3. **metric-style wide**
4. **segment-routing** mpls
5. **exit**
6. **show isis segment-routing**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**router isis** **Example:**	Enables the IS-IS routing protocol and enters router configuration mode.

	Command or Action	Purpose
	`Device(config-router)# router isis`	
Step 2	**net network-entity-title** **Example:** `Device(config-router)# net 49.0000.0000.0003.00`	Configures network entity titles (NETs) for the routing instance.
Step 3	**metric-style wide** **Example:** `Device(config-router)# metric-style wide`	Configures the device to generate and accept only wide link metrics.
Step 4	**segment-routing** mpls **Example:** `Device(config-router)# segment-routing mpls`	Configures segment routing operation state.
Step 5	**exit** **Example:** `Device(config-router)# exit`	Exits segment routing mode and returns to the configuration terminal mode.
Step 6	**show isis segment-routing** **Example:** `Device# show is-is segment-routing`	Displays the current state of the IS-IS segment routing.

Example

The following example displays output from the **show isis segment-routing state** command for the segment routing under IS-IS:

```
Device# show isis segment-routing

ISIS protocol is registered with MFI
ISIS MFI Client ID:0x63
Tag 1 - Segment-Routing:
   SR State:SR_ENABLED
   Number of SRGB:1
   SRGB Start:16000, Range:8000, srgb_handle:0x4500AED0, srgb_state: created
   Address-family IPv4 unicast SR is configured
     Operational state:Enabled
```

Configuring Prefix-SID for IS-IS

This section explains how to configure prefix segment identifier (SID) index under each interface.

Before you begin

Segment routing must be enabled on the corresponding address family.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **segment-routing mpls**
4. **connected-prefix-sid-map**
5. **address-family ipv4**
6. **10.1.1.1/32 index 100 range 1**
7. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**segment-routing mpls** **Example:** `Device(config)# segment-routing mpls`	Configures segment routing mpls mode.
Step 4	**connected-prefix-sid-map** **Example:** `Device(config-srmpls)# connected-prefix-sid-map`	Enters a sub-mode where you can configure address-family specific mappings for local prefixes and SIDs.
Step 5	**address-family ipv4** **Example:** `Device(config-srmpls-conn)# address-family ipv4`	Specifies the IPv4 address family and enters router address family configuration mode.
Step 6	**10.1.1.1/32 index 100 range 1** **Example:**	Associates SID 100 with the address 10.1.1.1/32.

	Command or Action	Purpose
	`Device(config-srmpls-conn-af)# 10.1.1.1/32 100 range 1`	
Step 7	**exit** **Example:** `Device(config-router)# exit`	Exits segment routing mode and returns to the configuration terminal mode.

Configuring Prefix Attribute N-Flag

By default, a flag called N-flag is set by IS-IS when advertising an SID that is associated with a loopback address. To clear this flag add explicit configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback3**
4. **isis prefix n-flag-clear**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback3** **Example:** `Device(config)# interface loopback3`	Specifies the interface loopback.
Step 4	**isis prefix n-flag-clear** **Example:** `Device(config-if)# isis prefix n-flag-clear`	Clears the prefix N-flag.

Configuring the Explicit Null Attribute

To disable penultimate-hop-popping (PHP) and add explicit-Null label, explicit-null option needs to be specified. Once the option is given, IS-IS sets the E flag in the prefix-SID sub TLV.

By default, a flag called E-flag (Explicit-Null flag) is set to 0 by ISIS when advertising a Prefix SID which is associated with a loopback address. If you wish to set this flag add explicit configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **segment-routing mpls**
4. **set-attributes**
5. **address-family ipv4**
6. **explicit-null**
7. **exit-address-family**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**segment-routing mpls** **Example:** `Device(config)# segment-routing mpls`	Configures segment routing mpls mode.
Step 4	**set-attributes** **Example:** `Device(config-srmpls)# set-attributes`	Sets the attribute.
Step 5	**address-family ipv4** **Example:** `Device(config-srmpls-attr)# address-family ipv4`	Specifies the IPv4 address family and enters router address family configuration mode.

	Command or Action	Purpose
Step 6	**explicit-null** **Example:** `Device(config-srmpls-attr-af)# explicit-null`	Enables the explicit-null label.
Step 7	**exit-address-family** **Example:** `Device(config-srmpls-attr-af)# exit-address-family`	Exits the address family.

Configuring Segment Routing Label Distribution Protocol Preference

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **segment-routing mpls**
4. **set-attributes**
5. **address-family ipv4**
6. **sr-label-preferred**
7. **exit-address-family**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**segment-routing mpls** **Example:** `Device(config)# segment-routing mpls`	Configures segment routing mpls mode.
Step 4	**set-attributes** **Example:**	Sets the attribute.

	Command or Action	Purpose
	`Device(config-srmpls)# set-attributes`	
Step 5	**address-family ipv4** **Example:** `Device(config-srmpls-attr)# address-family ipv4`	Specifies the IPv4 address family and enters router address family configuration mode.
Step 6	**sr-label-preferred** **Example:** `Device(config-srmpls-attr-af)# sr-label-preferred`	Specifies SR label to be preferred over the LDP.
Step 7	**exit-address-family** **Example:** `Device(config-srmpls-attr-af)# exit-address-family`	Exits the address family.

Configuring IS-IS SRMS

The following command enables the IS-IS SRMS and allows IS-IS to advertise local mapping entries. IS-IS does not send remote entries to the SRMS library. However, IS-IS uses the SRMS active policy, which is computed based only on the locally configured mapping entries.

```
[no] segment-routing prefix-sid-map advertise-local
```

Configuring IS-IS SRMS Client

By default, the IS-IS SRMS client mode is enabled. IS-IS always sends remote prefix-sid-mapping entries received through LSP to SRMS. The SRMS active policy is calculated based on local and remote mapping entries.

The following command disables the prefix-sid-mapping client functionality and it is configured on the receiver side:

```
segment-routing prefix-sid-map receive [disable]
```

Configuring IS-IS SID Binding TLV Domain Flooding

By default, the IS-IS SRMS server does not flood SID binding entries within the routing domain. From Cisco IOS-XE Release 3.18, the optional keyword **domain-wide** is added in the IS-IS SRMS server mode command to enable the SID and Label binding TLV flooding functionality:

```
segment-routing prefix-sid-map advertise-local [domain-wide]
```

The **domain-wide** keyword enables the IS-IS SRMS server to advertise SID binding TLV across the entire routing domain.

Note The option is valid only if IS-IS SRMS performs in the SRMS server mode.

Configuration Examples for Segment Routing —IS-IS v4 Node SID

Example: Configuring Segment Routing on IS-IS Network

The following example shows how to configure prefix segment identifier (SID) index under each interface:

```
Device(config)#segment-routing mpls
 Device(config-srmpls)#connected-prefix-sid-map
  Device(config-srmpls-conn)#address-family ipv4
   Device(config-srmpls-conn-af)#10.1.2.2/32 index 2 range 1
  Device(config-srmpls-conn-af)#exit-address-family
 Device(config-srmpls-conn-af)#end
```

Example: Configuring an Explicit Null Attribute

The following is an example of configuring an explicit null attribute:

```
Device(config)# segment-routing mpls
Device(config-srmpls)# set-attributes
 Device(config-srmpls-attr)# address-family ipv4
  Device(config-srmpls-attr-af)# explicit-null
 Device (config-srmpls-attr-af)# exit-address-family
```

Additional References for Segment Routing with IS-IS v4 Node SID

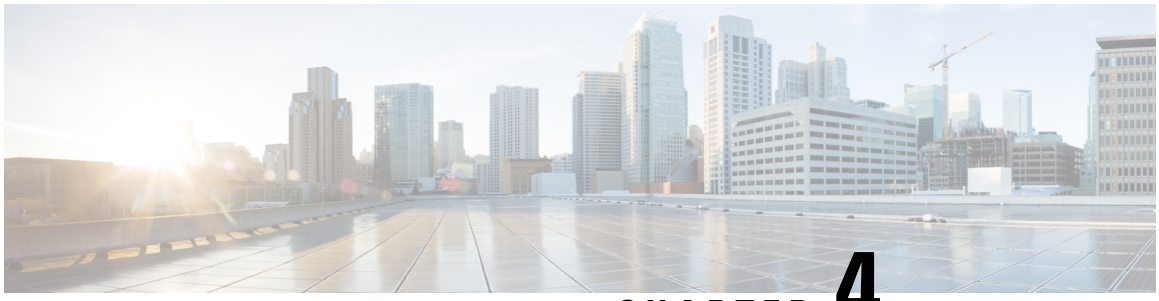
Related Documents

Related Topic	Document Title
IP Routing ISIS commands	Cisco IOS IP Routing ISIS commands

RFCs

RFC	Title
RFC4971	Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information
RFC5305	IS-IS Extensions for Traffic Engineering. Defines the advertisement of router IDs for IPv4.

RFC	Title
RFC6119	IPv6 Traffic Engineering in IS-IS. Defines the advertisement of router IDs for IPv6.



CHAPTER 4

IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute

This document describes the functionalities and IS-IS implementation of IP Fast Re-Route feature (IPFRR) using Segment Routing (SR) Topology Independent Loop Free Alternative (TI-LFA) link protection.

- Feature Information for IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute, on page 25
- Prerequisites for IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute, on page 26
- Restrictions for IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute , on page 27
- Information About IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute, on page 27
- How to Configure IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute, on page 30

Feature Information for IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute

Table 2: Feature Information for IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute

Feature Name	Releases	Feature Information
IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute	Cisco IOS XE Amsterdam 17.3.2	The following commands were introduced or modified: **fast-reroute ti-lfa {level-1 | level-2} [maximum-metric value]** , isis fast-reroute ti-lfa protection level-1 disable, **isis fast-reroute ti-lfa protection {level-1 | level-2} [maximum-metric value]**, **show running all | section interface interface-name**, **show running all | section router isis**.

Feature Name	Releases	Feature Information
IPv4 GRE-TP tunnel as protected link support for SR-MPLS TI-LFA with IS-IS	Cisco IOS XE 17.18.2	From Cisco IOS XE 17.18.2 this feature extends IPv4 GRE-TP tunnel as protected link support for SR-MPLS TILFA with ISIS.

Prerequisites for IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute

- Enable TI-LFA on all the nodes, before configuring SR-TE for TI-LFA.

```
mpls traffic-eng tunnels
!
segment-routing mpls
 connected-prefix-sid-map
  address-family ipv4
   10.1.1.1/32 index 11 range 1
  exit-address-family
 !
interface Loopback1
 ip address 10.1.1.1 255.255.255.255
 ip router isis 1
!
interface Tunnel1
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 10 explicit name IP_PATH segment-routing
!
interface GigabitEthernet2
 ip address 192.168.1.1 255.255.255.0
 ip router isis 1
 negotiation auto
 mpls traffic-eng tunnels
 isis network point-to-point
!
interface GigabitEthernet3
 ip address 192.168.2.1 255.255.255.0
 ip router isis 1
 negotiation auto
 mpls traffic-eng tunnels
 isis network point-to-point
!
router isis 1
 net 49.0001.0010.0100.1001.00
 is-type level-1
 metric-style wide
 log-adjacency-changes
 segment-routing mpls
 fast-reroute per-prefix level-1 all
 fast-reroute ti-lfa level-1
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng level-1
!
ip explicit-path name IP_PATH enable
 next-address 10.4.4.4
```

```
 next-address 10.5.5.5
 next-address 10.6.6.6
```

- If a microloop gets created between routers in case of primary and secondary path switch over you need to bring down the convergence time. Use the **microloop avoidance rib-update-delay** command to bring down the convergence time:

```
router isis ipfrr
net 49.0001.0120.1201.2012.00
is-type level-2-only
metric-style wide
log-adjacency-changes
segment-routing mpls
segment-routing prefix-sid-map advertise-local
fast-reroute per-prefix level-2 all
fast-reroute ti-lfa level-2
microloop avoidance rib-update-delay 10000
```

- Enable MPLS-TE nonstop routing (NSR) and IS-IS nonstop forwarding (NSF) to reduce or minimize traffic loss after a high availability (HA) switch over. Use the **mpls traffic-eng nsr** command in global exec mode.

```
mpls traffic-eng nsr
```

Use the **nsf** command under IS-IS.

```
router isis
nsf cisco
nsf interval 0
```

Restrictions for IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute

- The IGP cannot guarantee loop-free paths when tunnels are used for protection or repair, as it is unaware of the tunnel's underlying physical path.
- Forwarding failures on tunnel interfaces can occur in cases where local link failures do not occur. For this reason, in order to quickly detect forwarding failures on tunnel interfaces it is advisable to use the BFD protocol.

  Note that aggressive BFD timers must be used with care on tunnel interfaces as they can lead to false indication of failures
- Due to limitations on this platform, IGP TI-LFA cannot guarantee 50ms FRR failover.

Information About IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute

When the local LFA and remote LFA are enabled, there is a good coverage of the prefixes to be protected. However, for some rare topologies that do not have a PQ intersect node, both local and remote LFA will fail

to find a release node to protect the failed link. Furthermore, there is no way to prefer a post-convergence path, as the two algorithms have no knowledge of the post-convergence characteristics of the LFA.

To overcome the above limitation, effective Cisco IOS-XE Release 3.18, topology-independent LFA (TI-LFA) is supported on an SR-enabled network.

Topology-Independent Loop Free Alternate

TI-LFA provides supports for the following:

- Link Protection—The LFA provides repair path for failure of the link.
- Local LFA—Whenever a local LFA on the post convergence path is available, it is preferred over TI-LFA because local LFA does not require additional SID for the repair path. That is, the label for the PQ node is not needed for the release node.
- Local LFA for extended P space—For nodes in the extended P space, local LFA is still the most economical method for the repair path. In this case, TI-LFA will not be chosen.
- Tunnel to PQ intersect node—This is similar to remote LFA except that the repair path is guaranteed on the post convergence path using TI-LFA.
- Tunnel to PQ disjoint node—This capability is unique to the TI-LFA in the case when local and remote LFA cannot find a repair path.
- Tunnel to traverse multiple intersect or disjoint PQ nodes, up to the platform's maximum supported labels—TI-LFA provides complete coverage of all prefixes.
- P2P interfaces for the protected link—TI-LFA protects P2P interfaces.
- Asymmetrical links—The ISIS metrics between the neighbors are not the same.
- Multi-homed (anycast) prefix protection—The same prefix may be originated by multiple nodes.
- Protected prefix filtering—The route-map includes or excludes a list of prefixes to be protected and the option to limit the maximum repair distance to the release node.
- Tiebreakers—A subset of existing tiebreakers, applicable to TI-LFA, is supported.

Topology Independent Loop Free Alternate Tie-break

Local and remote LFA use default or user-configured heuristics to break the tie when there is more than one path to protect the prefix. The attributes are used to trim down the number of repair paths at the end of the TI-LFA link protection computation before the load balancing. Local LFA and remote LFA support the following tiebreakers:

- Linecard-disjoint—Prefers the line card disjoint repair path
- Lowest-backup-path-metric—Prefers the repair path with lowest total metric
- Node-protecting—Prefers node protecting repair path
- SRLG-disjoint—Prefers SRLG disjoint repair path
- Load-sharing—Distributes repair paths equally among links and prefixes

When there are two repair paths for a particular prefix, the path that the output port on different line card than that of the primary port is chosen as the repair path. For TI-LFA link protection, the following tiebreakers are supported:

- Linecard-disjoint—Prefers the line card disjoint repair path.
- LC disjoint index—If both the repair paths are on the same line card as that of the primary path, then, both paths are considered as candidates. If one of the path is on a different line card, then that path is chosen as the repair path.

- SRLG index—If both the repair paths have the same SRLG ID as that of the primary path, then, both the paths are considered as candidates. If one of the path has a different srlg id, then path is chosen as the repair path.
- Node-protecting—For TI-LFA node protection, the protected node is removed when computing the post-convergence shortest path. The repair path must direct traffic around the protected node.

The SRLG ID can be configured for each interface. When there are two repair paths for a prefix, the configured SRLG ID for the repair path is compared with that of the primary path SRLG ID. If the SRLG IDs for the secondary path is different than that of the primary, that path is chosen as the repair path. This policy comes into effect only when the primary path is configured with an SRLG ID. It is possible to configure both node and SRLG protection modes for the same interface or the same protocol instance. In that case, an additional TI-LFA node-SRLG combination protection algorithm is run. The TI-LFA node-SRLG combination algorithm removes the protected node and all members of the interface with the same SRLG group when computing the post-convergence SPT.

Interface Fast Reroute Tiebreakers

Interface fast reroute (FRR) tiebreakers are also needed for TI-LFA node and SRLG protection. When interface and protocol instance FRR tiebreakers both are configured, the interface FRR tiebreakers take precedence over the protocol instance. When interface FRR tiebreakers are not configured, the interface inherits the protocol instance FRR tiebreakers.

The following interface FRR tiebreaker commands apply only to the particular interface.

```
isis fast-reroute tie-break
[level-1 | level-2] linecard-disjoint
priority
isis fast-reroute tie-break
[level-1 | level-2] lowest-backup-metric
priority
isis fast-reroute tie-break
[level-1 | level-2] node-protecting
priority
isis fast-reroute tie-break
[level-1 | level-2] srlg-disjoint
priority
isis fast-reroute tie-break
[level-1 | level-2] default
```

Tie-breaker default and explicit tie-breaker on the same interface are mutually exclusive.

The following tie-breakers are enabled by default on all LFAs:

- linecard-disjoint
- lowest-backup-metric
- srlg-disjoint

Effective with Cisco IOS-XE Release 3.18, node-protecting tie-breaker is disabled by default.

How to Configure IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute

Perform the following steps to configure Link-protection Topology Independent Loop Free Alternate Fast Reroute.

Configuring Topology Independent Loop Free Alternate Fast Reroute

You can enable TI-LFA using any of the following two methods:

1. **Protocol enablement**—Enables TI-LFA in router isis mode for all IS-IS interfaces. Optionally, use the interface command to exclude the interfaces on which TI-LFA should be disabled.

   For example, to enable TI-LFA for all IS-IS interfaces:

   ```
   router isis 1
   fast-reroute per-prefix {level-1 | level-2}
   fast-reroute ti-lfa {level-1 | level-2} [maximum-metric value]
   ```

Note The **isis fast-reroute protection level-x** command enables local LFA and is required to enable TI-LFA.

2. **Interface enablement**—Enable TI-LFA selectively on each interface.

   ```
   interface interface-name
   isis fast-reroute protection {level-1 | level-2}
   isis fast-reroute ti-lfa protection {level-1 | level-2} [maximum-metric value]
   ```

   The **maximum-metric** option specifies the maximum repair distance which a node is still considered eligible as a release node.

   When both interface and protocol are TI-LFA enabled, the interface configuration takes precedence over the protocol configuration. TI-LFA is disabled by default.

   To disable TI-LFA on a particular interface, use the following command:

   ```
   interface interface-name
   isis fast-reroute ti-lfa protection level-1 disable
   ```

Configuring Topology Independent Loop Free Alternate With Mapping Server

Consider the following topology to understand the configuration:



- IXIA-2 injects ISIS prefixes, and IXIA-1 sends one-way traffic to IXIA-2

  .
- In R1 10,000 prefixes are configured in the segment-routing mapping-server.

The configuration on R1 is:

```
configure terminal
segment-routing mpls
global-block 16 20016
!
connected-prefix-sid-map
address-family ipv4
10.11.11.11/32 index 11 range 1
exit-address-family
!
!
mapping-server
!
prefix-sid-map
address-family ipv4
10.0.0.0/24 index 2 range 1 attach
203.0.113.1/24 index 1 range 1 attach
192.168.0.0/24 index 100 range 10000 attach
exit-address-family
!
!
!
!
interface Loopback0
ip address 10.11.11.11 255.255.255.255
ip router isis ipfrr
!
interface GigabitEthernet0/1/0
ip address 10.14.0.1 255.255.255.0
ip router isis ipfrr
negotiation auto
isis network point-to-point
!
interface GigabitEthernet0/1/2
ip address 10.0.0.1 255.255.255.0
ip router isis ipfrr
negotiation auto
isis network point-to-point
!
interface GigabitEthernet0/1/4
ip address 203.0.113.2 255.255.255.0
ip router isis ipfrr
negotiation auto
isis network point-to-point
!
router isis ipfrr
net 49.0001.0110.1101.1011.00
is-type level-2-only
metric-style wide
log-adjacency-changes
nsf cisco
segment-routing mpls
segment-routing prefix-sid-map advertise-local
fast-reroute per-prefix level-2 all
fast-reroute ti-lfa level-2
microloop avoidance rib-update-delay 10000
```

On R2 the configuration is

```
configure terminal
!
!
segment-routing mpls
!
connected-prefix-sid-map
address-family ipv4
10.12.12.12/32 index 12 range 1
exit-address-family
!
!
interface Loopback0
ip address 10.12.12.12 255.255.255.255
ip router isis ipfrr
!
interface GigabitEthernet0/1/0
ip address 10.12.0.1 255.255.255.0
ip router isis ipfrr
negotiation auto
isis network point-to-point
!
interface GigabitEthernet0/1/1
ip address 10.11.0.2 255.255.255.0
ip router isis ipfrr
negotiation auto
isis network point-to-point
!
router isis ipfrr
net 49.0001.0120.1201.2012.00
is-type level-2-only
metric-style wide
log-adjacency-changes
nsf cisco
segment-routing mpls
segment-routing prefix-sid-map advertise-local
fast-reroute per-prefix level-2 all
fast-reroute ti-lfa level-2
microloop avoidance rib-update-delay 10000
!
```

On R3 the configuration is

```
configure terminal
!
mpls traffic-eng tunnels
!
segment-routing mpls
!
connected-prefix-sid-map
address-family ipv4
10.13.13.13/32 index 13 range 1
exit-address-family
!
!
interface Loopback0
ip address 10.13.13.13 255.255.255.255
ip router isis ipfrr
!
interface GigabitEthernet0/0/4
ip address 10.13.0.1 255.255.255.0
ip router isis ipfrr
```

```
load-interval 30
speed 1000
no negotiation auto
isis network point-to-point
!
interface GigabitEthernet0/0/5
ip address 10.12.0.2 255.255.255.0
ip router isis ipfrr
negotiation auto
isis network point-to-point
!
router isis ipfrr
net 49.0001.0130.1301.3013.00
is-type level-2-only
metric-style wide
log-adjacency-changes
nsf cisco
segment-routing mpls
segment-routing prefix-sid-map advertise-local
fast-reroute per-prefix level-2 all
fast-reroute ti-lfa level-2
microloop avoidance rib-update-delay 10000
!
```

On R4 the configuration is:

```
configure terminal
!
mpls traffic-eng tunnels
!
segment-routing mpls
!
connected-prefix-sid-map
address-family ipv4
10.14.14.14/32 index 14 range 1
exit-address-family
!
!
interface Loopback0
ip address 10.14.14.14 255.255.255.255
ip router isis ipfrr
!
interface GigabitEthernet0/0/0
ip address 10.14.0.2 255.255.255.0
ip router isis ipfrr
negotiation auto
isis network point-to-point
!
interface GigabitEthernet0/0/3
ip address 10.13.0.2 255.255.255.0
ip router isis ipfrr
speed 1000
no negotiation auto
isis network point-to-point
!
interface GigabitEthernet0/0/5
ip address 10.120.0.1 255.255.255.0
ip router isis ipfrr
speed 1000
no negotiation auto
isis network point-to-point
!
router isis ipfrr
net 49.0001.0140.1401.4014.00
```

```
is-type level-2-only
metric-style wide
log-adjacency-changes
nsf cisco
segment-routing mpls
segment-routing prefix-sid-map advertise-local
fast-reroute per-prefix level-2 all
fast-reroute ti-lfa level-2
microloop avoidance rib-update-delay 10000
!
```

Examples: Configuring IS-IS Link-protection Topology Independent Loop Free Alternate Fast Reroute

Example 1: In the following example, local LFA is configured with linecard-disjoint and srlg-disjoint tiebreakers. Linecard-disjoint is given preference with a lower priority value (10) than the srlg-disjoint (11).

```
router isis access
 net 49.0001.2037.0685.b002.00
 metric-style wide
 fast-flood 10
 max-lsp-lifetime 65535
 lsp-refresh-interval 65000
 spf-interval 5 50 200
 prc-interval 5 50 200
 lsp-gen-interval 5 5 200
 log-adjacency-changes
 nsf ietf
 segment-routing mpls
 fast-reroute per-prefix level-1 all – configures the local LFA
 fast-reroute per-prefix level-2 all
 fast-reroute remote-lfa level-1 mpls-ldp - enables rLFA (optional)
 fast-reroute remote-lfa level-2 mpls-ldp
 fast-reroute ti-lfa level-1 - enables TI-LFA
 microloop avoidance rib-update-delay 10000
 bfd all-interfaces
```

Example 2—Enable TI-LFA node-protecting tie-breaker on all ISIS level-2 interfaces with priority 100. All other tiebreakers are disabled.

```
router isis
fast-reroute per-prefix level-2 all
fast-reroute ti-lfa level-2
fast-reroute tie-break level-2 node-protecting 100
```

Example 3—Enable TI-LFA node-protecting tie-breaker with priority 100 and TI-LFA SRLG protection with priority 200 on all IS-IS level-2 interfaces. All other tiebreakers are disabled because the node-protecting tie-breaker is configured.

```
router isis
fast-reroute per-prefix level-2 all
fast-reroute ti-lfa level-2
fast-reroute tie-break level-2 node-protecting 100
fast-reroute tie-break level-2 srlg-disjoint 200
```

Example 4—Enable TI-LFA node-protecting tie-breaker with priority 100 on all ISIS level-2 interfaces except on Ethernet0/0. For those IS-IS interfaces, all other tiebreakers are disabled. Ethernet0/0 overwrites the

inheritance and uses the default set of tiebreakers with linecard-disjoint, lowest-backup-path-metric, srlg-disjoint enabled.

```
router isis
fast-reroute per-prefix level-2 all
fast-reroute ti-lfa level-2
fast-reroute tie-break level-2 node-protecting 100
!
interface ethernet0/0
ip router isis
isis fast-reroute tie-break level-2 default
```

Example 5—Enable TI-LFA using the default tiebreaker on all IS-IS interfaces except on Ethernet0/0. On Ethernet0/0 enable TI-LFA node-protecting with priority 100 and disable all other tiebreakers.

```
router isis
fast-reroute per-prefix level-2 all
fast-reroute ti-lfa level-2
!
interface ethernet0/0
ip router isis
isis fast-reroute tie-break level-2 node-protecting 100
```

Example 6—Enable TI-LFA node-protecting tie-breaker with priority 200 and linecard-disjoint tie-breaker with priority 100 on all ISIS level-2 interfaces. All other tiebreakers are disabled.

```
router isis
fast-reroute per-prefix level-2 all
fast-reroute ti-lfa level-2
fast-reroute tie-break level-2 linecard-disjoint 100
fast-reroute tie-break level-2 node-protecting 200
```

Verifying the Tiebreaker

To view tiebreakers enabled on the interface, use the following command:

show running all | section interface *interface-name*

To view tiebreakers enabled on the router mode, use the following command:

show running all | section router isis

Verifying the Primary and Repair Paths

In this example, 10.1.1.1 is the protecting neighbor and 10.4.4.4 is the neighbor on the protecting link.

```
Router#
show ip cef 10.1.1.1
10.1.1.1/32
  nexthop 10.1.1.1 GigabitEthernet0/2/0 label [explicit-null|explicit-null]() - slot 2 is
primary interface
    repair: attached-nexthop 10.24.0.2 TenGigabitEthernet0/3/0 - slot 3 is repair interface

  nexthop 10.24.0.2 TenGigabitEthernet0/3/0 label [explicit-null|explicit-null]()
    repair: attached-nexthop 10.1.1.1 GigabitEthernet0/2/0
Router#
show ip cef 10.4.4.4
10.4.4.4/32
```

```
  nexthop 10.4.4.4 GigabitEthernet0/2/3 label [explicit-null|16004]() - slot 2 is primary
interface
    repair: attached-nexthop 10.5.5.5 MPLS-SR-Tunnel2
Router# show ip cef 10.4.4.4 int
10.4.4.4/32, epoch 3, RIB[I], refcnt 6, per-destination sharing
  sources: RIB, Adj, LTE
  feature space:
    IPRM: 0x00028000
    Broker: linked, distributed at 4th priority
    LFD: 10.4.4.4/32 2 local labels
    dflt local label info: global/877 [0x3]
    sr local label info: global/16004 [0x1B]
        contains path extension list
        dflt disposition chain 0x46654200
          label implicit-null
          FRR Primary
            <primary: IP adj out of GigabitEthernet0/2/3, addr 10.4.4.4>
        dflt label switch chain 0x46654268
          label implicit-null
          TAG adj out of GigabitEthernet0/2/3, addr 10.4.4.4
        sr disposition chain 0x46654880
          label explicit-null
          FRR Primary
            <primary: TAG adj out of GigabitEthernet0/2/3, addr 10.4.4.4>
        sr label switch chain 0x46654880
          label explicit-null
          FRR Primary
            <primary: TAG adj out of GigabitEthernet0/2/3, addr 10.4.4.4>
  subblocks:
    Adj source: IP adj out of GigabitEthernet0/2/3, addr 10.4.4.4 464C6620
      Dependent covered prefix type adjfib, cover 10.0.0.0/0
  ifnums:
    GigabitEthernet0/2/3(11): 10.4.4.4
    MPLS-SR-Tunnel2(1022)
  path list 3B1FC930, 15 locks, per-destination, flags 0x4D [shble, hvsh, rif, hwcn]
    path 3C04D5E0, share 1/1, type attached nexthop, for IPv4, flags [has-rpr]
     MPLS short path extensions: [rib | lblmrg | srlbl] MOI flags = 0x21 label explicit-null

      nexthop 10.4.4.4 GigabitEthernet0/2/3 label [explicit-null|16004](), IP adj out of
GigabitEthernet0/2/3, addr 10.4.4.4 464C6620
        repair: attached-nexthop 10.5.5.5 MPLS-SR-Tunnel2 (3C04D6B0)
    path 3C04D6B0, share 1/1, type attached nexthop, for IPv4, flags [rpr, rpr-only]
      MPLS short path extensions: [rib | lblmrg | srlbl] MOI flags = 0x1 label 16004
      nexthop 10.5.5.5 MPLS-SR-Tunnel2 label 16004(), repair, IP midchain out of
MPLS-SR-Tunnel2 46CE2440
  output chain:
    label [explicit-null|16004]()
    FRR Primary (0x3B209220)
      <primary: TAG adj out of GigabitEthernet0/2/3, addr 10.4.4.4 464C6480> - primary path

      <repair:  TAG midchain out of MPLS-SR-Tunnel2 46CE22A0
                label 16()
                label 16003()
                TAG adj out of TenGigabitEthernet0/3/0, addr 10.24.0.2 46CE25E0> - repair
path
```

Verifying the IS-IS Segment Routing Configuration

```
Router# show isis segment-routing
 ISIS protocol is registered with MFI
 ISIS MFI Client ID:0x63
 Tag Null - Segment-Routing:
```

```
SR State:SR_ENABLED
Number of SRGB:1
SRGB Start:14000, Range:1001, srgb_handle:0xE0934788, srgb_state: created
Address-family IPv4 unicast SR is configured
  Operational state: Enabled
```

The command with keyword **global-block** displays the SRGB and the range for LSPs.

```
Router# show isis segment-routing global-block
IS-IS Level-1 Segment-routing Global Blocks:
System ID             SRGB Base    SRGB Range
nevada                20000        4001
arizona             * 16000        1000
utah                  40000        8000
```

The **show isis segment-routing prefix-sid-map** command with keyword **advertise** displays the prefix-sid maps that the router advertises.

```
Roouter# show isis segment-routing prefix-sid-map adv
IS-IS Level-1 advertise prefix-sid maps:
Prefix               SID Index    Range        Flags
10.16.16.16/32       101          1
10.16.16.17/32       102          1            Attached
```

The **show isis segment-routing prefix-sid-map** command with keyword **receive** displays the prefix-sid maps that the router receives.

```
Router #sh isis segment-routing prefix-sid-map receive
IS-IS Level-1 receive prefix-sid maps:
Host                 Prefix               SID Index    Range        Flags
utah                 10.16.16.16/32       101          1
                     10.16.16.17/32       102          1            Attached
```

To display the connected-SIDs found in the LSPs and passed to the mapping server component, use the **show isis segment-routing connected-sid** command.

```
Router# show isis segment-routing connected-sid
IS-IS Level-1 connected-sids
Host                  Prefix               SID Index    Range        Flags
nevada              * 10.1.1.2/32          1002         1
                      10.2.2.2/32          20           1
                      10.1.1.10/32         10           1
colorado              10.1.1.3/32          33           1
                      10.1.1.6/32          6            1
IS-IS Level-2 connected-sids
Host                 Prefix               SID Index    Range        Flags
```

Verifying the IS-IS Topology Independent Loop Free Alternate Tunnels

```
Router# show isis fast-reroute ti-lfa tunnel
Fast-Reroute TI-LFA Tunnels:
Tunnel  Interface  Next Hop         End Point        Label      End Point Host
MP1     Et1/0      10.30.1.4        10.1.1.2          41002     nevada
MP2     Et0/0      10.19.1.6        10.1.1.6          60006     colorado
                                    10.1.1.2          16        nevada
MP3     Et0/0      10.19.1.6        10.1.1.6          60006     colorado
                                    10.1.1.2          16        nevada
                                    10.1.1.5          70005     wyoming
```

Verifying the Segment Routing Traffic Engineering With Topology Independent Loop Free Alternate Configuration

```
Router# show mpls traffic-eng tunnels tunnel1
Name: PE1                              (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type explicit IP_PATH (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 0 [0] bw-based
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 4 hours, 25 minutes
      Time since path change: 4 hours, 21 minutes
      Number of LSP IDs (Tun_Instances) used: 37
    Current LSP: [ID: 37]
      Uptime: 4 hours, 21 minutes
  Tun_Instance: 37
  Segment-Routing Path Info (isis  level-1)
    Segment0[Node]: 10.4.4.4, Label: 16014
    Segment1[Node]: 10.5.5.5, Label: 16015
    Segment2[Node]: 10.6.6.6, Label: 16016
Router# show isis fast-reroute ti-lfa tunnel

Tag 1:
Fast-Reroute TI-LFA Tunnels:
Tunnel  Interface  Next Hop         End Point        Label     End Point Host
MP1     Gi2        192.168.1.2      10.6.6.6         16016     SR_R6
MP2     Gi3        192.168.2.2      10.6.6.6         16016     SR_R6
Router# show frr-manager client client-name ISIS interfaces detail
TunnelI/F : MP1
  Type : SR
  Next-hop : 192.168.1.2
  End-point : 10.6.6.6
  OutI/F : Gi2
  Adjacency State : 1
  Prefix0 : 10.6.6.6(Label : 16016)
TunnelI/F : MP2
  Type : SR
  Next-hop : 192.168.2.2
  End-point : 10.6.6.6
  OutI/F : Gi3
  Adjacency State : 1
  Prefix0 : 10.6.6.6(Label : 16016)
Router# show ip cef 10.6.6.6 internal

10.6.6.6/32, epoch 2, RIB[I], refcnt 6, per-destination sharing
  sources: RIB, LTE
  feature space:
    IPRM: 0x00028000
    Broker: linked, distributed at 1st priority
```

```
    LFD: 10.6.6.6/32 1 local label
    sr local label info: global/16016 [0x1A]
        contains path extension list
        sr disposition chain 0x7FC6B0BF2AF0
          label implicit-null
          IP midchain out of Tunnel1
          label 16016
          FRR Primary
            <primary: label 16015
                      TAG adj out of GigabitEthernet3, addr 192.168.2.2>
        sr label switch chain 0x7FC6B0BF2B88
          label implicit-null
          TAG midchain out of Tunnel1
          label 16016
          FRR Primary
            <primary: label 16015
                      TAG adj out of GigabitEthernet3, addr 192.168.2.2>
  ifnums:
    Tunnel1(13)
  path list 7FC6B0BBDDE0, 3 locks, per-destination, flags 0x49 [shble, rif, hwcn]
    path 7FC7144D4300, share 1/1, type attached nexthop, for IPv4
      MPLS short path extensions: [rib | prfmfi | lblmrg | srlbl] MOI flags = 0x3 label
implicit-null
      nexthop 10.6.6.6 Tunnel1, IP midchain out of Tunnel1 7FC6B0BBB440
  output chain:
    IP midchain out of Tunnel1 7FC6B0BBB440
    label [16016|16016]
    FRR Primary (0x7FC714515460)
      <primary: label 16015
                TAG adj out of GigabitEthernet3, addr 192.168.2.2 7FC6B0BBB630>
      <repair:  label 16015
                label 16014
                TAG midchain out of MPLS-SR-Tunnel1 7FC6B0BBAA90
                label 16016
                TAG adj out of GigabitEthernet2, addr 192.168.1.2 7FC6B0BBBA10>
```

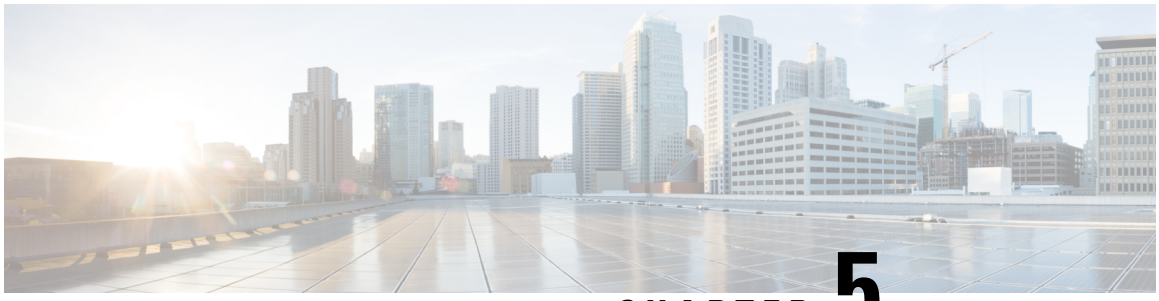
Note To ensure a less than 50 msec traffic protection with TI-LFA, SR-TE with dynamic path option must use the backup adjacency SID.

To create an SR-TE with dynamic path option, use the following configuration on every router in the topology:

```
router isis 1
fast-reroute per-prefix level-1 all
```

At the tunnel head-end router:

```
interface Tunnel1
ip unnumbered Loopback1
tunnel mode mpls traffic-eng
tunnel destination 10.6.6.6
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 1 dynamic segment-routing
tunnel mpls traffic-eng path-selection segment-routing adjacency protected
```



CHAPTER 5

Segment Routing Traffic Engineering With IS-IS

This chapter describes how segment routing traffic engineering (SR-TE) can be implemented using IS-IS and contains the following sections:

- Feature Information for Segment Routing Traffic Engineering with IS-IS, on page 41
- Restrictions for Segment Routing-Traffic Engineering with IS-IS, on page 42
- Information About Segment Routing Traffic Engineering with IS-IS, on page 42
- How to Configure Segment Routing Traffic Engineering with IS-IS, on page 48

Feature Information for Segment Routing Traffic Engineering with IS-IS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 3: Feature Information for Segment Routing Traffic Engineering with IS-IS

Feature Name	Releases	Feature Information
Segment Routing Traffic Engineering with IS-IS	Cisco IOS XE Amsterdam 17.3.2	The following commands were introduced or modified: • **mpls traffic-eng nsr** • **show mpls traffic-eng tunnels tunnel1** • **show isis fast-reroute ti-lfa tunnel** • **show frr-manager client client-name ISIS interfaces detail** • **show ip cef 6.6.6.6 internal**

Restrictions for Segment Routing-Traffic Engineering with IS-IS

- SR-TE is not supported on broadcast interfaces; it is supported only point-to-point interfaces.
- SR-MPLS policies do not support an IPv6 GRE tunnel as the first hop. If a selected path uses an IPv6 GRE tunnel, first-hop resolution fails and the policy remains operationally down. Use an IPv4 GRE tunnel for the SR-MPLS path.
- Only one instance of protocol should be enabled for TE at a given point of time.
- You can use the verbatim keyword only on a label-switched path (LSP) that is configured with the explicit path option.
- Re-optimization is unsupported on the verbatim LSP.

Information About Segment Routing Traffic Engineering with IS-IS

A Traffic Engineered (TE) tunnel is a container of TE LSPs instantiated between the tunnel ingress and the tunnel destination. A TE tunnel can instantiate one or more SR-TE LSPs that are associated with the same tunnel. The SR-TE LSP path may not necessarily follow the same IGP path to a destination node. In this case, the SR-TE path can be specified through either a set of prefix-SIDs, or adjacency-SIDs of nodes, or both, and links to be traversed by the SR-TE LSP.

The head-end imposes the corresponding MPLS label stack on outgoing packets to be carried over the tunnel. Each transit node along the SR-TE LSP path uses the incoming top label to select the next-hop, pop or swap the label, and forward the packet to the next node with the remainder of the label stack, until the packet reaches the ultimate destination. The set of hops or segments that define an SR-TE LSP path are provisioned by the operator.

SR-TE LSP Instantiation

A Traffic Engineered (TE) tunnel is a container of one or more instantiated TE LSPs. An SR-TE LSP is instantiated by configuring ‘segment-routing’ on the path-option of the TE tunnel. The traffic mapped to the tunnel is forwarded over the primary SR-TE instantiated LSP.

Multiple path-options can also be configured under the same tunnel. Each path-option is assigned a preference index or a path-option index that is used to determine the more favorable path-option for instantiating the primary LSP—the lower the path-option preference index, the more favorable the path-option. The other less favorable path-options under the same TE tunnel are considered secondary path-options and may be used once the currently used path-option is invalidated (for example, due to a failure on the path.



Note A forwarding state is maintained for the primary LSP only.

SR-TE LSP Explicit Null

MPLS-TE tunnel head-end does not impose explicit-null at the bottom of the stack. When penultimate hop popping (PHP) is enabled for SR prefix SIDs or when an adjacency SID is the last hop of the SR-TE LSP, the packet may arrive at the tail-end without a transport label. However, in some cases, it is desirable that the packet arrive at the tail-end with explicit-null label, and in such case, the head-end will impose an explicit-null label at the top of the label stack.

SR-TE LSP Path Verification

SR-TE tunnel functionality requires that the head-end perform initial verification of the tunnel path as well as the subsequent tracking of the reachability of the tunnel tail-end and traversed segments.

Path verification for SR-TE LSP paths is triggered whenever MPLS-TE is notified of any topology changes or SR SID updates.

The SR-TE LSP validation steps consist of the following checks:

Topology Path Validation

The head-end validates the path of an SR-TE LSP for connectivity against the TE topology. MPLS-TE head-end checks if links corresponding to the adjacency SIDs are connected in the TE topology.

For newly-instantiated SR-TE LSPs, if the head-end detects a discontinuity on any link of the SR-TE path, that path is considered invalid and is not used. If the tunnel has other path-options with valid paths, those paths are used to instantiate the tunnel LSP.

For TE tunnels with existing instantiated SR-TE LSP, if the head-end detects a discontinuity on any link, the head-end assumes a fault has occurred on that link. In this case, the local repair protection, such as the IP FRR, come in to effect. The IGPs continue to sustain the protected adjacency label and associated forwarding after the adjacency is lost for some time. This allows the head-ends enough time to reroute the tunnels onto different paths that are not affected by the same failure. The head-end starts a tunnel invalidation timer once it detects the link failure to attempt to reroute the tunnel onto other available path-options with valid paths.

If the TE tunnel is configured with other path-options that are not affected by the failure and are validated, the head-end uses one of those path-options to reroute (and re-optimize) the tunnel by instantiating a new primary LSP for the tunnel using the unaffected path.

If no other valid path-options exist under the same tunnel, or if the TE tunnel is configured with only one path-option that is affected by the failure, the head-end starts an invalidation timer after which it brings the tunnel state to ‘down’. This action avoids a null route from being sent along with traffic flowing over the affected SR-TE LSP, and allows services riding over the tunnel to reroute over different available paths at the head-end. There is an invalidation drop configuration that keeps the tunnel 'up', but drops the traffic when the invalidation timer expires.

For intra-area SR-TE LSPs, the head-end has full visibility over the LSP path, and validates the path to the ultimate LSP destination. However, for inter-area LSPs, the head-end has partial visibility over the LSP path—only up to the first ABR. In this case, the head-end can only validate the path from the ingress to the first ABR. Any failure along the LSP beyond the first ABR node is invisible to the head-end, and other mechanisms to detect such failures, such as BFD over LSP are assumed.

SR SID Validation

SID hops of an SR-TE LSP are used to determine the outgoing MPLS label stack to be imposed on the outgoing packets carried over the SR-TE LSP of a TE tunnel. A database of global and local adjacency-SIDs is populated from the information received from IGPs and maintained in MPLS-TE. Using a SID that is not available in

the MPLS TE database invalidates the path-option using the explicit-path. The path-option, in this case, is not used to instantiate the SR TE LSP. Also, withdrawing, adding, or modifying a SID in the MPLS-TE SID-database, results in the MPLS-TE head-end verifying all tunnels with SR path-options (in-use or secondary) and invokes proper handling.

LSP Egress Interface

When the SR-TE LSP uses an adjacency-SID for the first path hop, TE monitors the interface state and IGP adjacency state associated with the adjacency-SID and the node that the SR-TE LSP egresses on. If the interface or adjacency goes down, TE can assume a fault occurred on the SR-TE LSP path and take the same reactive actions described in the previous sections.

Note When the SR-TE LSP uses a prefix-SID for the first hop, TE cannot directly infer on which interface the tunnel egresses. TE relies on the IP reachability information of the prefix to determine if connectivity to the first hop is maintained.

IP Reachability Validation

MPLS-TE validates that the nodes corresponding to the prefix-SIDs are IP reachable before declaring the SR path valid. MPLS-TE detects path changes for the IP prefixes corresponding to the adjacency or prefix SIDs of the SR-TE LSP path. If the node announcing a specific SID loses IP reachability. due to a link or node failure, MPLS-TE is notified of the path change (no path). MPLS-TE reacts by invalidating the current SR-TE LSP path, and may use other path-options with a valid path, if any to instantiate a new SR-TE LSP.

Note Since IP-FRR does not offer protection against failure of a node that is being traversed by an SR-TE LSP (such as, a prefix-SID failure along the SR-TE LSP path), the head-end immediately reacts to IP route reachability loss for prefix-SID node by setting the tunnel state to ‘down’ and removes the tunnel forwarding entry if there are no other path-options with valid path for the affected tunnel.

Tunnel Path Affinity Validation

The affinity of a tunnel path can be specified using the command **tunnel mpls traffic-eng affinity** under the tunnel interface.

The head-end validates that the specified SR path is compliant with the configured affinity. This necessitates that the paths of each segment of the SR path be validated against the specified constraint. The path is declared invalid against the configured affinity constraints if at least a single segment of the path does not satisfy the configured affinity.

Tunnel Path Resource Avoidance Validation

You can specify a set of addresses to be validated as excluded from being traversed by SR-TE tunnel packets. To achieve this, the head-end runs the per-segment verification checks and validates that the specified node, prefix or link addresses are indeed excluded from the tunnel in the SR path. The tunnel resource avoidance checks can be enabled per path using the commands below. The list of addresses to be excluded are defined and the name of the list is referenced in the path-option.

```
interface tunnel100
 tunnel mpls traffic-eng path-option 1 explicit name EXCLUDE segment-routing
```

```
ip explicit-path name EXCLUDE enable
 exclude-address 192.168.0.2
 exclude-address 192.168.0.4
 exclude-address 192.168.0.3
!
```

Verbatim Path Support

MPLS TE LSPs usually require that all the nodes in the network are TE aware which means that they have IGP extensions to TE in place. However, some network administrators want the ability to build TE LSPs to traverse nodes that do not support IGP extensions to TE, but that do support RSVP extensions to TE. Verbatim LSPs are helpful when all or some of the intermediate nodes in a network do not support IGP extensions for TE.

When this feature is enabled, the IP explicit path is not checked against the TE topology database. Since the TE topology database is not verified, a Path message with IP explicit path information is routed using the shortest path first (SPF) algorithm for IP routing.

SR-TE Traffic Load Balancing

SR-TE tunnels support the following load-balancing options:

Load Balancing on Port Channel TE Links

Port Channel interfaces carry the SR-TE LSP traffic. This traffic load balances over port channel member links as well as over bundle interfaces on the head or mid of an SR-TE LSP.

Load Balancing on Single Tunnel

While using the equal cost multi path protocol (ECMP), the path to a specific prefix-SID may point to multiple next-hops. And if the SR-TE LSP path traverses one or more prefix-SIDs that have ECMP, the SR-TE LSP traffic load-balances on the ECMP paths of each traversed prefix-SID from the head-end or any midpoint traversed node along the SR-TE LSP path.

Load Balancing on Multiple Tunnels

Multiple TE tunnels can be used as next-hop paths for routes to specific IP prefixes either by configuring static route on multiple tunnels, or auto-route announcing multiple parallel tunnels to the same destination. In such cases, the tunnels share the traffic load equally or load balance traffic on multiple parallel tunnels. It is also possible to allow Unequal Load Balance (UELB) with an explicit per tunnel configuration at the tunnel head-end. In this case, the tunnel load-share is passed from MPLS-TE to forwarding plane.

The tunnel load-share feature continues to work for TE tunnels that instantiate the SR-TE LSPs.

SR-TE Tunnel Reoptimization

TE tunnel reoptimization occurs when the head-end determines that there is a more optimal path available than the one currently used. For example, in case of a failure along the SR-TE LSP path, the head-end could detect and revert to a more optimal path by triggering reoptimization.

Tunnels that instantiate SR-TE LSP can re-optimize without affecting the traffic carried over the tunnel.

Re-optimization can occur because:

- The explicit path hops used by the primary SR-TE LSP explicit path are modified.
- The head-end determines the currently used path-option are invalid due to either a topology path disconnect, or a missing SID in the SID database that is specified in the explicit-path.

- A more favorable path-option (lower index) becomes available.

When the head-end detects a failure on a protected SR adjacency-SID that is traversed by an SR-TE LSP, it starts the invalidation timer. If the timer expires and the head-end is still using the failed path because it is unable to reroute on a different path, the tunnel state is brought 'down' to avoid a null route from being sent along with the traffic. Once the tunnel is down, services on the tunnel converge to take a different path.

The following is a sample output of a manual reoptimization example. In this example, the path-option is changed from **10** to **20**.

```
Router# mpls traffic-eng reoptimize tunnel 1 path-option 20
The targeted path-option is not in lock down mode. Continue? [no]: yes
Router# show mpls traffic-eng tunnels tunnel1
Name: R1_t1                           (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 20, (SEGMENT-ROUTING) type explicit IP_PATH (Basis for Setup)
    path option 10, (SEGMENT-ROUTING) type dynamic
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 20 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours, 9 minutes
      Time since path change: 14 seconds
      Number of LSP IDs (Tun_Instances) used: 1819
    Current LSP: [ID: 1819]
      Uptime: 17 seconds
      Selection: reoptimization
    Prior LSP: [ID: 1818]
      ID: path option unknown
      Removal Trigger: reoptimization completed
  Tun_Instance: 1819
  Segment-Routing Path Info (isis level-1)
    Segment0[Node]: 10.4.4.4, Label: 114
    Segment1[Node]: 10.5.5.5, Label: 115
    Segment2[Node]: 10.6.6.6, Label: 116
```

SR-TE with Lockdown Option

The **lockdown** option prevents SR-TE from re-optimizing to a better path. However, it does not prevent signaling the existence of a new path.

```
interface Tunnel1
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng path-option 10 segment-routing lockdown
 tunnel mpls traffic-eng path-selection metric igp
```

```
 tunnel mpls traffic-eng load-share 10                          (Tunnel1) Destination:
10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (LOCKDOWN) type segment-routing (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: enabled  Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: segment-routing path option 10 is active
    BandwidthOverride: disabled  LockDown: enabled   Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours, 22 minutes
      Time since path change: 1 minutes, 26 seconds
      Number of LSP IDs (Tun_Instances) used: 1822
    Current LSP: [ID: 1822]
      Uptime: 1 minutes, 26 seconds
      Selection: reoptimization
    Prior LSP: [ID: 1821]
      ID: path option unknown
      Removal Trigger: configuration changed
  Tun_Instance: 1822
  Segment-Routing Path Info (isis  level-1)
    Segment0[Node]: 10.6.6.6, Label: 116
```

SR-TE Tunnel Protection

Protection for SR TE tunnels can take any of the following alternatives:

IP-FRR Local Repair Protection

On an SR-TE LSP head-end or mid-point node, IP-FRR is used to compute and program the backup protection path for the prefix-SID or adjacency-SID label.

With IP-FRR, backup repair paths are pre-computed and pre-programmed by IGPs *before* a link or node failure. The failure of a link triggers its immediate withdrawal from the TE topology (link advertisement withdrawal). This allows the head-end to detect the failure of an SR-TE LSP traversing the failed adjacency-SID.

When a protected adjacency-SID fails, the failed adjacency-SID label and associated forwarding are kept functional for a specified period of time (5 to 15 minutes) to allow all SR TE tunnel head-ends to detect and react to the failure. Traffic using the adjacency-SID label continues to be FRR protected even if there are subsequent topology updates that change the backup repair path. In this case, the IGPs update the backup repair path while FRR is active to reroute traffic on the newly-computed backup path.

When the primary path of a protected prefix-SID fails, the PLR reroutes to the backup path. The head-end remains transparent to the failure and continues to use the SR-TE LSP as a valid path.

IP-FRR provides protection for adjacency and prefix-SIDs against link failures only.

Tunnel Path Protection

Path protection is the instantiation of one or more standby LSPs to protect against the failure of the primary LSP of a single TE tunnel.

Path protection protects against failures by pre-computing and pre-provisioning secondary paths that are failure diverse with the primary path-option under the same tunnel. This protection is achieved by computing a path that excludes prefix-SIDs and adjacency-SIDs traversed by the primary LSP or by computing a path that excludes SRLGs of the primary SR-TE LSP path.

In the event of a failure of the primary SR-TE LSP, at least one standby SR-TE LSP is used for the tunnel. Multiple secondary path-options can be configured to be used as standby SR-TE LSPs paths.

Unnumbered Support

IS-IS description of an unnumbered link does not contain remote interface ID information. The remote interface ID of an unnumbered link is required to include the unnumbered link as part of the SR-TE tunnel.

How to Configure Segment Routing Traffic Engineering with IS-IS

Perform the following steps to configure Segment Routing Traffic Engineering (SR-TE) with IS-IS.

Configuring the Path Option for a TE Tunnel

When the path-option type for an operational SR tunnel is changed from SR to non-SR (for example, **dynamic**), the existing forwarding entry of the tunnel is deleted.

Segment Routing can be enabled or disabled on an existing secondary or an in-use path-option. If the tunnel uses a signaled RSVP-TE explicit path-option and segment routing is enabled on that tunnel, the RSVP-TE LSP is torn, and the SR-TE LSP is instantiated using the same path-option. Conversely, if segment routing is disabled on a path-option that is in use by the primary LSP, the tunnel goes down intermittently and a new RSVP-TE LSP will be signaled using the same explicit path.

If the segment-routing path-option is enabled on a secondary path-option (that is, not in-use by the tunnel's primary LSP), the tunnel is checked to evaluate if the newly specified SR-TE LSP path-option is valid and more favorable to use for the tunnel primary LSP.

```
Device(config)# interface tunnel 100
Device(config-if)# tunnel mpls traffic-eng path-option 1 explicit name foo segment-routing
Device(config-if)# tunnel mpls traffic-eng path-option 2 dynamic segment-routing
Device(config-if)# tunnel mpls traffic-eng path-option 3 segment-routing
```

Configuring SR Explicit Path Hops

The following explicit path hops are supported in SR-TE:

- IP addresses
- MPLS labels
- Mix of IP addresses and MPLS labels

For intra-area LSPs, the explicit path can be specified as a list of IP addresses:

```
Device(config)# ip explicit-path name foo
Device(config-ip-expl-path)# index 10 next-address 10.1.1.1 node address
Device(config-ip-expl-path)# index 20 next-address 10.12.12.2 link address
```

Note When using IP unnumbered interfaces, you cannot specify next hop address as an explicit path index. It should be node address or label.

The explicit path can also be specified as segment-routing SIDs:

```
Device(config)# ip explicit-path name foo
Device(config-ip-expl-path)# index 10 next-label 20
```

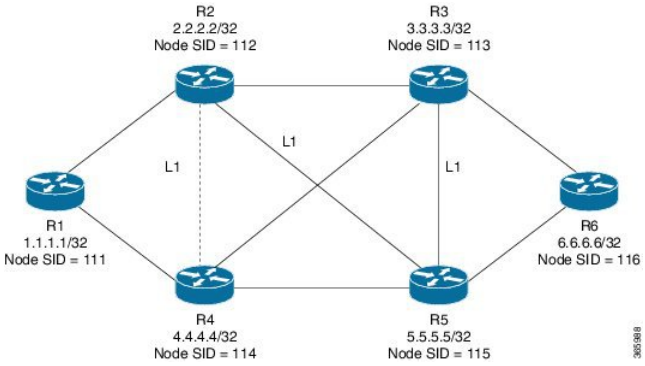
Configuring Affinity on an Interface

Perform the following steps to configure affinity on an interface:

```
interface GigabitEthernet2
 ip address 192.168.2.1 255.255.255.0
 ip router isis 1
 negotiation auto
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 isis network point-to-point
 ip rsvp bandwidth
```

Use Case: Segment Routing Traffic Engineering Basic Configuration

Consider the following topology to understand the SR-TE configuration:



To configure at the head-end router, R1:

```
!
mpls traffic-eng tunnels
!
segment-routing mpls
connected-prefix-sid-map
  address-family ipv4
   10.1.1.1/32 index 111  range 1
  exit-address-family
```

```
!
set-attributes
  address-family ipv4
  sr-label-preferred
exit-address-family
!
interface Loopback1
ip address 10.1.1.1 255.255.255.255
ip router isis 1
!
int gig0/0
ip address 10.11.11.1 255.255.255.0
ip router isis 1
mpls traffic-eng tunnels
isis network point-to-point
!
router isis 1
net 49.0001.0010.0100.1001.00
is-type level-1
metric-style wide
segment-routing mpls
segment-routing prefix-sid-map advertise-local
mpls traffic-eng router-id  Loopback1
mpls traffic-eng level-1
!
end
```

To enable SR-TE Explicit path (Node SID based), enable the following CLI on R1:

```
Head end SR-TE configuration R1#
!
interface tunnel1
ip unnumbered Loopback1
tunnel mode mpls traffic-eng
tunnel destination 10.6.6.6
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 6 6
tunnel mpls traffic-eng path-option 10 explicit name Node_PATH segment-routing
!
ip explicit-path name Node_PATH
 next-label  16114
next-label  16115
next-label  16116
```

To verify proper operation of SR-TE tunnel 1 on R1 enable the following CLI:

```
Tunnel verification on (R1)# show mpls traffic-eng tun tun 1 detail
Name: R1_t1                             (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type explicit Node_PATH (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
Verbatim: disabled
Number of LSP IDs (Tun_Instances) used: 1815
    Current LSP: [ID: 1815]
      Uptime: 2 seconds
Removal Trigger: configuration changed
    Segment-Routing Path Info (isis  level-1)
    Segment0[Node]: 10.4.4.4, Label: 16114
```

```
    Segment1[Node]: 10.5.5.5, Label: 16115
    Segment2[Node]: 10.6.6.6, Label: 16116
```

To configure at the tail-end router, R6:

```
interface GigabitEthernet2
ip address 10.101.1.1 255.255.255.0
ip router isis 1
negotiation auto
mpls traffic-eng tunnels
router isis 1
 net 49.0001.0060.0600.6006.00
 ispf level-1
 metric-style wide
 log-adjacency-changes
 segment-routing mpls

segment-routing prefix-sid-map advertise-local
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng level-1
```

Explicit Path SR-TE Tunnel 1

Consider tunnel 1 based only on IP addresses:

```
ip explicit-path name IP_PATH1
 next-address 10.2.2.2
 next-address 10.3.3.3
 next-address 10.6.6.6
!
interface Tunnel1
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng path-option 10 explicit name IP_PATH1 segment-routing
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10
end
```

Explicit Path SR-TE Tunnel 2

Consider tunnel 2 based on node SIDs

```
ip explicit-path name IA_PATH
 next-label 114
 next-label 115
 next-label 116
!
interface Tunnel2
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng bandwidth 10000 class-type 1
 tunnel mpls traffic-eng path-option 10 explicit name IA_PATH segment-routing
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10
end
```

Explicit Path SR-TE Tunnel 3

Consider that tunnel 3 is based on a mix of IP addresses and label

```
ip explicit-path name MIXED_PATH enable
 next-address 10.2.2.2
 next-address 10.3.3.3
 next-label 115
 next-label 116
!
interface Tunnel3
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng path-option 10 explicit name MIXED_PATH segment-routing
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10
```



Note In the case of mixed path, IP next-hop cannot be used after using Node SIDs in the path. The following path will not be valid:

```
ip explicit-path name MIXED_PATH enable
next-label 115
next-label 116
next-address 10.2.2.2
```

Dynamic Path SR-TE Tunnel 4

Consider that tunnel 4is based on adjacency SIDs

```
interface Tunnel4
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng bandwidth 10000 class-type 1
 tunnel mpls traffic-eng path-option 10 dynamic segment-routing
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10
end
```

Dynamic Path SR-TE Tunnel 5

Consider that tunnel 5 is based on Node SIDs

```
 interface Tunnel5
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng path-option 10 segment-routing
```

```
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10
```

Verifying Configuration of the SR-TE Tunnels

Use the **show mpls traffic-eng tunnels** *tunnel-number* command to verify the configuration of the SR-TE tunnels.

Verifying Tunnel 1

```
Name: R1_t1                             (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type explicit IP_PATH (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours
      Time since path change: 2 seconds
      Number of LSP IDs (Tun_Instances) used: 1814
    Current LSP: [ID: 1814]
      Uptime: 2 seconds
      Selection: reoptimization
    Prior LSP: [ID: 1813]
      ID: path option unknown
      Removal Trigger: configuration changed
  Tun_Instance: 1814
  Segment-Routing Path Info (isis  level-1)
    Segment0[Node]: 10.4.4.4, Label: 114
    Segment1[Node]: 10.5.5.5, Label: 115
    Segment2[Node]: 10.6.6.6, Label: 116
```

Verifying Tunnel 2

```
Name: R1_t2                             (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type explicit IA_PATH (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
```

```
    State: explicit path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours, 1 minutes
      Time since path change: 1 seconds
      Number of LSP IDs (Tun_Instances) used: 1815
    Current LSP: [ID: 1815]
      Uptime: 1 seconds
    Prior LSP: [ID: 1814]
      ID: path option unknown
      Removal Trigger: configuration changed
  Tun_Instance: 1815
  Segment-Routing Path Info (isis  level-1)
    Segment0[ - ]: Label: 114
    Segment1[ - ]: Label: 115
    Segment2[ - ]: Label: 116
```

Verifying Tunnel 3

```
Name: R1_t3                             (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type explicit MIXED_PATH (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours, 2 minutes
      Time since path change: 2 seconds
      Number of LSP IDs (Tun_Instances) used: 1816
    Current LSP: [ID: 1816]
      Uptime: 2 seconds
      Selection: reoptimization
    Prior LSP: [ID: 1815]
      ID: path option unknown
      Removal Trigger: configuration changed
  Tun_Instance: 1816
  Segment-Routing Path Info (isis  level-1)
    Segment0[Node]: 10.2.2.2, Label: 112
    Segment1[Node]: 10.3.3.3, Label: 113
    Segment2[ - ]: Label: 115
    Segment3[ - ]: Label: 116
```

Verifying Tunnel 4

```
Name: R1_t4                             (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type dynamic (Basis for Setup, path weight 30)
  Config Parameters:
```

```
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: dynamic path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours
      Time since path change: 2 seconds
      Number of LSP IDs (Tun_Instances) used: 1813
    Current LSP: [ID: 1813]
      Uptime: 2 seconds
    Prior LSP: [ID: 1806]
      ID: path option unknown
      Removal Trigger: configuration changed
  Tun_Instance: 1813
  Segment-Routing Path Info (isis  level-1)
    Segment0[Link]: 192.168.2.1 - 192.168.2.2, Label: 17
    Segment1[Link]: 192.168.4.2 - 192.168.4.1, Label: 25
    Segment2[Link]: 192.168.8.1 - 192.168.8.2, Label: 300
```

Verifying Tunnel 5

```
Name: R1_t5                             (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, type segment-routing (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: segment-routing path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours, 4 minutes
      Time since path change: 14 seconds
      Number of LSP IDs (Tun_Instances) used: 1817
    Current LSP: [ID: 1817]
      Uptime: 14 seconds
      Selection: reoptimization
    Prior LSP: [ID: 1816]
      ID: path option unknown
      Removal Trigger: configuration changed
  Tun_Instance: 1817
  Segment-Routing Path Info (isis  level-1)
    Segment0[Node]: 10.6.6.6, Label: 116
```

Verifying Verbatim Path Support

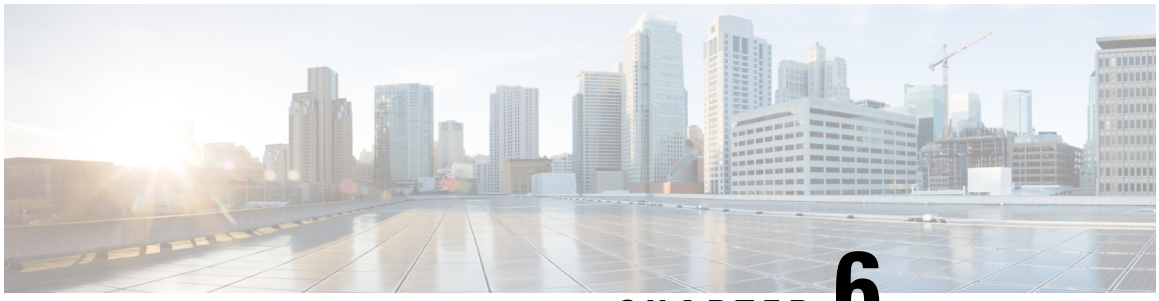
To verify proper operation and SR-TE tunnel state use following CLI:

```
R6#sh mpl traffic-eng tunnels tunnel 4

Name: R6_t4                               (Tunnel4) Destination: 10.11.11.11
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, (SEGMENT-ROUTING) type explicit (verbatim) multihop (Basis for Setup)

  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    Path Selection:
     Protection: any (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
    Hop Limit: disabled [ignore: Verbatim Path Option]
    Cost Limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear
    AutoRoute: disabled LockDown: disabled Loadshare: 0 [0] bw-based
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: enabled

  History:
    Tunnel:
      Time since created: 16 minutes, 40 seconds
      Time since path change: 13 minutes, 6 seconds
      Number of LSP IDs (Tun_Instances) used: 13
    Current LSP: [ID: 13]
      Uptime: 13 minutes, 6 seconds
      Selection: reoptimization
    Prior LSP: [ID: 12]
      ID: path option unknown
      Removal Trigger: configuration changed (severe)
  Tun_Instance: 13
  Segment-Routing Path Info (IGP information is not used)
    Segment0[First Hop]: 10.0.0.0, Label: 16003
    Segment1[ - ]: Label: 16002
    Segment2[ - ]: Label: 16001
```



CHAPTER 6

Segment Routing With OSPFv2 Node SID

This chapter describes how Segment Routing works with OSPFv2 node SID.

- Feature Information for Segment Routing With OSPFv2 Node SID, on page 57
- Information About Segment Routing With OSPFv2 Node SID, on page 57
- How to Configure Segment Routing With OSPFv2 Node SID, on page 60
- Additional References for Segment Routing With OSPFv2 Node SID, on page 68

Feature Information for Segment Routing With OSPFv2 Node SID

Table 4: Feature Information for Segment Routing With OSPFv2 Node SID

Feature Name	Releases	Feature Information
Segment Routing With OSPF	Cisco IOS XE Amsterdam 17.3.2	The Segment Routing OSPFv2 node SID feature provides support for segment routing on OSPF networks. The following commands were introduced or modified: **connected-prefix-sid-map**, **show ip ospf 10 segment-routing**, **sr-label-preferred**, **ip ospf prefix-attributes n-flag-clear** .

Information About Segment Routing With OSPFv2 Node SID

Segment Routing relies on a small number of extensions to Open Shortest Path First (OSPF) protocols. There are two levels of configuration required to enable segment routing for a routing protocol instance. The top level segment routing configuration which is managed by segment routing infrastructure component enables segment routing, whereas, segment routing configuration at the router ospf level enables segment routing for the ospf instance. There are three segment routing states:

- SR_NOT_CONFIGURED
- SR_DISABLED
- SR_ENABLED

Segment routing configuration under the IGPs is allowed only if the SR state is either SR_DISABLED or SR_ENABLED. The SR_ENABLED state indicates that there is at least a valid SRGB range reserved. You can enable segment routing for IGPs under the router configuration sub mode, through commands. However, IGP segment routing are enabled only after the global SR is configured.

The SR_ENABLED is a necessary state for any protocol to enable SR, however, it is not a sufficient for enabling SR for a protocol instance. The reason being that the OSPF still does not have any information about segment routing global block (SRGB) information. When the request to receive information about the SRGB is processed successfully, the OSPF SR operational state is enabled.

Segment Routing requires each router to advertise its segment routing data-plane capability and the range of MPLS label values that are used for segment routing in the case where global SIDs are allocated. Data-plane capabilities and label ranges are advertised using the SR-capabilities sub-TLV inserted into the OSPF Router Information Opaque LSA.

OSPF SR-capabilities sub TLV includes all reserved SRGB ranges. However, the Cisco implementation supports only one SRGB range.

Prefix-SID Received in Label Switched Path From Remote Routers

OSPF sends the prefix SIDs associated with the connected prefix using the Extended Prefix Sub TLV in its opaque Extended prefix LSA. Prefix SIDs received in a LSA which have got reachability are downloaded to the routing information base (RIB) in the same way as BGP downloads per prefix VPN labels, only if the following conditions are met:

- Segment routing is enabled for the topology and address-family.
- Prefix-SID is valid.
- The local label binding to MFI is successful.



Note For SIDs that do not fit in the specified SID range, labels are not used when updating the RIB. For the cases, where SID fits in the SID range, but does not fit the next-hop neighbor SID range, remote label associated with that path is not installed.

Segment Routing Adjacency SID Advertisement

Effective with Cisco IOS-XE Release 3.17, OSPF supports the advertisement of segment routing adjacency SID. An Adjacency Segment Identifier (Adj-SID) represents a router adjacency in Segment Routing.

A segment routing-capable router may allocate an Adj-SID for each of its adjacencies and an Adj-SID sub-TLV is defined to carry this SID in the Extended Opaque Link LSA.

OSPF allocates the adjacency SID for each OSPF neighbor if the OSPF adjacency which are in two way or in FULL state. OSPF allocates the adjacency SID only if the Segment Routing is enabled. The label for adjacency SID is dynamically allocated by the system. This eliminates the chances of misconfiguration, as this has got only the local significance.

Multiple Adjacency-SIDs

Effective with Cisco IOS-XE Release 16.3, multiple adjacency-SIDs are supported. For each OSPF adjacency, OSPF allots to Adj SIDs, unprotected and protected Adj-SIDs which are carried in the extended link LSAs. The protected adjacency SID (or back up Adj-SID) is allocated and advertised only when FRR is enabled on the router and also on the interface where SR is enabled on the system. When FRR or SR is disabled, the protected Adj-SID is released.

The persistence of protected adj-SID in forwarding plane is supported. When the primary link is down, OSPF delays the release of its backup Adj-SID until the delay timer (30 sec) expires. This allows the forwarding plane to continue to forward the traffic through the backup path until the router is converged.

The allocated and advertised backup Adj-SIDs can be displayed in the output of **show ip ospf neighbor detail** and **show ip ospf segment-routing protected-adjacencies command**.

Segment Routing Mapping Server

Segment Routing Mapping Server (SRMS) allows configuration and maintenance of the Prefix-SID mapping policy entries. Effective with Cisco IOS-XE Release 3.17, the IGPs use the active policy of the SRMS to determine the SID values when programming the forwarding plane.

The SRMS provides prefixes to SID/Label mapping policy for the network. IGPs, on the other hand, are responsible for advertising prefixes to SID/Label mapping policy through the Prefix-SID/Label Binding TLV.

Active policy information and changes are notified to the IGPs, which use active policy information to update forwarding information.

Connected Prefix SIDs

When a router installs a prefix with a SID that is different than what it advertises to the LSP, for example, if more than one protocol or more than one IGP instance is announcing the same prefix with different SIDs to the SRMS, the SRMS resolves the conflict and announces the winning prefix and SID that may not be the same as the local instance. In that case, the IGP always advertises what it learns from its source LSP although it still tries to install the SID which may be different than what it learns in its LSP. This is done to prevent the IGP from redistributing the SIDs from another protocol or another protocol instance.

SRGB Range Changes

When OSPF segment routing is configured, OSPF must request an interaction with the SRGB before OSPF SR operational state can be enabled. If no SRGB range is created, OSPF will not be enabled.

When an SRGB change event occurs, OSPF makes the corresponding changes in its sub-block entries. OSPF also advertises the newly created or extended SRGB range in SR-capabilities sub-TLV and updates the prefix-sid sub TLV advertisement.

MPLS Forwarding on an Interface

MPLS forwarding must be enabled before segment routing can use an interface. OSPF is responsible for enabling MPLS forwarding on an interface.

When segment routing is enabled for a OSPF topology, or OSPF segment routing operational state is enabled, it enables MPLS for any interface on which the OSPF topology is active. Similarly, when segment routing is disabled for a OSPF topology, it disables the MPLS forwarding on all interfaces for that topology.

Conflict Handling of SID Entries

When there is a conflict between the SID entries and the associated prefix entries use any of the following methods to resolve the conflict:

- When the system receives two SID entries for the same prefix, then the prefix received by higher router ID is treated as the SID corresponding to the prefix. The prefix is installed with the SID entry which was advertised by the higher router ID.
- When the system receives two SID entries one by OSPF protocol and the other by IS-IS protocol, then the SID entry received by OSPF protocol is treated as valid SID. The prefix is installed with the SID entry which was received by OSPF protocol.
- When two prefixes are advertised with the same SID entry, then the prefix which is advertised by the higher router ID is installed with the SID entry and the other prefix is installed without any SID entry.

In an ideal situation, each prefix should have unique SID entries assigned.

How to Configure Segment Routing With OSPFv2 Node SID

Perform the following steps to configure segment routing with OSPFv2 node SID.

Configuring Segment Routing With OSPF

Before you begin

Before configuring OSPF to support segment routing you must first configure the segment routing feature in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. segment-routing mpls
4. connected-prefix-sid-map
5. address-family ipv4
6. 10.1.1.1/32 index 100 range 1
7. exit-address-family

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	segment-routing mpls **Example:** `Device(config-sr)# segment-routing mpls`	Enables the segment feature using the mpls data plane.
Step 4	connected-prefix-sid-map **Example:** `Device(config-srmpls)# connected-prefix-sid-map`	Enters a sub-mode where you can configure address-family specific mappings for local prefixes and SIDs.
Step 5	address-family ipv4 **Example:** `Device(config-srmpls-conn)# address-family ipv4`	Specifies IPv4 address prefixes.
Step 6	10.1.1.1/32 index 100 range 1 **Example:** `Device(config-srmpls-conn-af)# 10.1.1.1/32 100 range 1`	Associates SID 100 with the address 10.1.1.1/32.
Step 7	exit-address-family **Example:** `Device(config-srmpls-conn-af)# exit-address-family`	Exits the address family.

Configuring Segment Routing on OSPF Network

Before you begin

Before you configure segment routing on OSPF network, OSPF must be enabled on your network.

SUMMARY STEPS

1. **router ospf 10**
2. **router-id**<*id*>
3. **segment-routing mpls**
4. **segment-routing area** <*area id*> **mpls**
5. **show ip ospf 10 segment-routing**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**router ospf 10** **Example:** `Device(config)# router ospf 10`	Enables the OSPF mode.
Step 2	**router-id**<*id*> **Example:** `Device(config-router)# router-id 10.0.0.0`	Configures OSPF routes.
Step 3	**segment-routing mpls** **Example:** `Device(config-router)# segment-routing mpls`	Configures segment routing mpls mode.
Step 4	**segment-routing area** <*area id*> **mpls** **Example:** `Device(config-router) # segment-routing area 0 mpls`	Configures segment routing mpls mode in a specific area.

Step 5	**show ip ospf 10 segment-routing**
	Example:
	`Device# show ip ospf 10 segment-routing`
		Shows the output for configuring SR under OSPF.
		The following example displays output from the show ip ospf segment-routing state command for the segment routing under OSPF:
		`Device#show ip ospf 10 segment-routing`
		`OSPF Router with ID (10.0.0.1) (Process ID 10)`
		`Global segment-routing state: Enabled`
		`Segment Routing enabled:`
		`Area Topology name Forwarding`
		`0 Base MPLS`
		`1 Base MPLS`
		`SR Attributes`
		`Prefer non-SR (LDP) Labels`
		`Do not advertise Explicit Null`
		`Local MPLS label block (SRGB):`
		`Range: 16000 - 23999`
		`State: Created`
		`Registered with SR App, client handle: 3`
		`Connected map notifications active (handle 0x4),`

	Command or Action	Purpose
		bitmask 0x1 Active policy map notifications active (handle 0x5), bitmask 0xC Registered with MPLS, client-id: 100 Bind Retry timer not running Adj Label Bind Retry timer not running

Configuring Prefix-SID for OSPF

This task explains how to configure prefix segment identifier (SID) index under each interface.

Before you begin

Segment routing must be enabled on the corresponding address family.

SUMMARY STEPS

1. enable
2. configure terminal
3. segment-routing mpls
4. connected-prefix-sid-map
5. address-family ipv4
6. 10.1.1.1/32 index 100 range 1
7. exit

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device# enable`	Enables privileged EXEC mode.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	segment-routing mpls **Example:** `Device(config)# segment-routing mpls`	Configures segment routing mpls mode.

	Command or Action	Purpose
Step 4	connected-prefix-sid-map **Example:** `Device(config-srmpls)# connected-prefix-sid-map`	Enters a sub-mode where you can configure address-family specific mappings for local prefixes and SIDs.
Step 5	address-family ipv4 **Example:** `Device(config-srmpls-conn)# address-family ipv4`	Specifies the IPv4 address family and enters router address family configuration mode.
Step 6	10.1.1.1/32 index 100 range 1 **Example:** `Device(config-srmpls-conn-af)# 10.1.1.1/32 100 range 1`	Associates SID 100 with the address 10.1.1.1/32.
Step 7	exit **Example:** `Device(config-router)# exit`	Exits segment routing mode and returns to the configuration terminal mode.

Configuring Prefix Attribute N-flag-clear

OSPF advertises prefix SIDs via Extended Prefix TLV in its opaque LSAs. It carries flags for the prefix and one of them is N flag (Node) indicating that any traffic sent along to the prefix is destined to the router originating the LSA. This flag typically marks host routes of router's loopback.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface loopback3
4. ip ospf prefix-attributes n-flag-clear

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	interface loopback3 **Example:** `Device(config)# interface loopback3`	Specifies the interface loopback.
Step 4	ip ospf prefix-attributes n-flag-clear **Example:** `Device(config-if)# ip ospf prefix-attributes n-flag-clear`	Clears the prefix N-flag.

Configuring Explicit Null Attribute With OSPF

To disable penultimate-hop-popping (PHP) and add explicit-Null label, explicit-null option needs to be specified. Once the option is given, OSPF sets the E flag in the Extended prefix-SID TLV in its LSAs.

By default, a flag called E-flag (Explicit-Null flag) is set to 0 by OSPF when advertising a Prefix SID which is associated with a loopback address. If you wish to set this flag add explicit configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. segment-routing mpls
4. set-attributes
5. address-family ipv4
6. explicit-null
7. exit-address-family

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	segment-routing mpls **Example:** `Device(config)# segment-routing mpls`	Configures segment routing mpls mode.
Step 4	set-attributes **Example:** `Device(config-srmpls)# set-attributes`	Sets the attribute.
Step 5	address-family ipv4 **Example:** `Device(config-srmpls-attr)# address-family ipv4`	Specifies the IPv4 address family and enters router address family configuration mode.
Step 6	explicit-null **Example:** `Device(config-srmpls-attr-af)# explicit-null`	Specifies the explicit-null.
Step 7	exit-address-family **Example:** `Device(config-srmpls-attr-af)# exit-address-family`	Exits the address family.

Configuring Segment Routing Label Distribution Protocol Preference With OSPF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. segment-routing mpls
4. set-attributes
5. address-family ipv4
6. sr-label-preferred
7. exit-address-family

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	segment-routing mpls **Example:** `Device(config)# segment-routing mpls`	Configures segment routing mpls mode.
Step 4	set-attributes **Example:** `Device(config-srmpls)# set-attributes`	Sets the attribute.
Step 5	address-family ipv4 **Example:** `Device(config-srmpls-attr)# address-family ipv4`	Specifies the IPv4 address family and enters router address family configuration mode.
Step 6	sr-label-preferred **Example:** `Device(config-srmpls-attr-af)# sr-label-preferred`	Specifies SR label to be preferred over the LDP.
Step 7	exit-address-family **Example:** `Device(config-srmpls-attr-af)# exit-address-family`	Exits the address family.

Configuring OSPF SRMS

The following command enables the OSPF SRMS and allows OSPF to advertise local mapping entries. OSPF does not send remote entries to the SRMS library. However, OSPF uses the SRMS active policy, which is computed based only on the locally configured mapping entries.

```
[no] segment-routing prefix-sid-map advertise-local
```

Configuring OSPF SRMS Client

By default, the OSPF SRMS client mode is enabled. OSPF always sends remote prefix-sid-mapping entries received through LSAs, to SRMS. The SRMS active policy is calculated based on both, local and remote mapping entries.

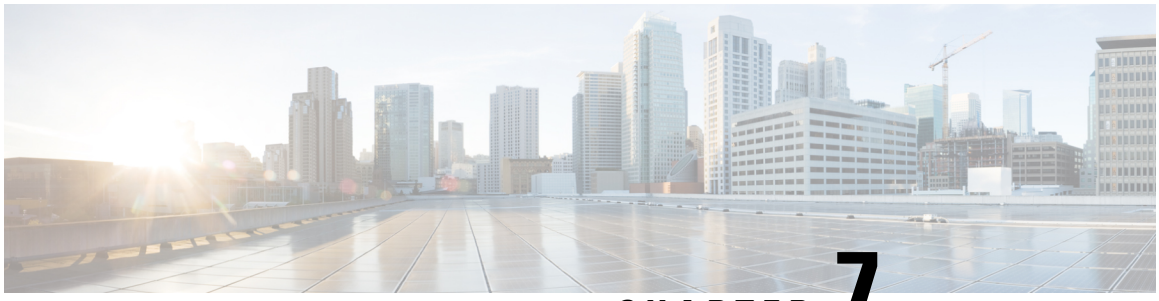
The following command disables the prefix-sid-mapping client functionality and it is configured on the receiver side.

```
segment-routing prefix-sid-map receive [disable]
```

Additional References for Segment Routing With OSPFv2 Node SID

Related Documents

Related Topic	Document Title
IP Routing ISIS commands	Cisco IOS IP Routing ISIS commands



CHAPTER 7

OSPFv2 Link-protection Topology Independent Loop Free Alternate Fast Reroute

This document describes OSPFv2 implementation of IP Fast Re-Route Feature (IP FRR) using TI -LFA (Topology Independent Loop Free Alternative).

- Feature Information for OSPFv2 Link-protection Topology Independent Loop Free Alternate Fast Reroute, on page 69
- Restrictions for Topology Independent Loop Free Alternate Fast Reroute, on page 70
- Information About OSPFv2 Link-Protection Topology Independent Loop Free Alternate Fast Reroute, on page 71
- How to Configure Topology Independent Loop Free Alternate Fast Reroute, on page 78
- Debugging Topology Independent Loop Free Alternate Fast Reroute, on page 83
- Examples: OSPFv2 Link-Protection Topology Independent Loop Free Alternate Fast Reroute, on page 83

Feature Information for OSPFv2 Link-protection Topology Independent Loop Free Alternate Fast Reroute

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 5: Feature Information for OSPFv2 Link-protection Topology Independent Loop Free Alternate Fast Reroute

Feature Name	Releases	Feature Information
OSPFv2 Link-protection Topology Independent Loop Free Alternate Fast Reroute	Cisco IOS XE Amsterdam 17.3.2	Topology-Independent Loop-Free Alternate (TI-LFA) uses segment routing to provide link, node, and Shared Risk Link Groups (SRLG) protection in topologies where other fast reroute techniques cannot provide protection. The goal of TI-LFA is to reduce the packet loss that results while routers converge after a topology change due to a link failure. The following commands were introduced or modified: **fast-reroute per-prefix ti-lfa [area <area> [disable]]**, **fast-reroute per-prefix tie-break node-protecting index <index>**, **fast-reroute per-prefix tie-break node-protecting required index <index>**, **fast-reroute per-prefix tie-break srlg index <index>**, **fast-reroute per-prefix tie-break srlg required index <index>**, **ip ospf fast-reroute per-prefix protection disable**, **ip ospf fast-reroute per-prefix candidate disable**, **show ip ospf fast-reroute ti-lfa tunnels**.
IPv4 GRE-TP tunnel as protected link support for SR-MPLS TI-LFA with OSPFv2	Cisco IOS XE 17.18.2	From Cisco IOS XE 17.18.2 this feature extends IPv4 GRE-TP tunnel as protected link support for SR-MPLS TILFA with OSPFv2.

Restrictions for Topology Independent Loop Free Alternate Fast Reroute

- TI-LFA is supported only on OSPFv2.
- TI-LFA tunnels are created only if the router supports SR and it is configured with prefix SID. The prefix (or) node SID can be configured as connected SID (or) advertised using the SRMS (Segment Routing Mapping Server).
- TI-LFA is not supported on OSPF point to multi point interfaces.
- TI-LFA does not support Multi Topology Routing (MTR).
- TI-LFA does not create the repair path using virtual link, sham link (or) TE tunnels.
- TI-LFA tunnel is constructed and programmed by explicitly specifying the node (or) set of repair nodes through which the tunnel needs to traverse.
- The IGP cannot guarantee loop-free paths when tunnels are used for protection or repair, as it is unaware of the tunnel's underlying physical path.
- Forwarding failures on tunnel interfaces can occur in cases where local link failures do not occur. For this reason, in order to quickly detect forwarding failures on tunnel interfaces it is advisable to use the BFD protocol.

Note that aggressive BFD timers must be used with care on tunnel interfaces as they can lead to false indication of failures

- Due to limitations on this platform, IGP TI-LFA cannot guarantee 50ms FRR failover.

Information About OSPFv2 Link-Protection Topology Independent Loop Free Alternate Fast Reroute

Topology-Independent Loop-Free Alternate (TI-LFA) uses segment routing to provide link, node, and Shared Risk Link Groups (SRLG) protection in topologies where other fast reroute techniques, such as RLFA (Remote Loop Free Alternative) cannot provide protection. The goal of TI-LFA is to reduce the packet loss that results while routers converge after a topology change due to a link failure.

These are the major benefits of using TI-LFA:

- Provides 100% coverage for all the prefixes.
- Prevents transient congestion and sub-optimal routing by leveraging on the post-convergence path.
- Protects Label Distribution Protocol (LDP) and IP traffic as well.

IP Fast Reroute and Remote Loop Free Alternate

IP Fast Reroute (FRR) is a set of techniques that allow rerouting the IP traffic around a failed link or failed node in the network.One of the techniques that is used is Loop Free Alternates (LFA), which is implemented using OSPF protocol. OSPF currently supports per-prefix directly connected LFA and remote LFA (RLFA). The problem with these LFA algorithms is the topology dependency; the LFA algorithms cannot find a loop-free alternate path through the network for all the topologies.

The per-prefix directly connected LFA (also known as DLFA) provides loop-free alternate path for most triangular topologies, but does not provide good coverage for rectangular or circular topologies. The Remote LFA implementation (RLFA) which uses MPLS forwarding with LDP signaling for tunneling the rerouted traffic to an intermediate node, extends the IPFRR coverage in ring or rectangular topologies. For each link, RLFA defines P-Space (set of nodes reachable from calculating node without crossing the protected link) and Q-Space (set of nodes that can reach the neighbor on the protected link without crossing the protected link itself). The nodes that belong to both P and Q-Spaces are called PQ nodes and can be used as the intermediate node for the protected traffic. RLFA forms targeted LDP session to the PQ node and form the RLFA tunnel. But for the topologies where P and Q-Spaces are disjoint, R-LFA does not provide protection for those prefixes.

Topology Independent Fast Reroute

Topology Independent Fast Reroute (TI-FRR) is a technique which uses segment routing to provide link protection in any topology assuming the metric on the links in the topology is symmetrical. TI-LFA does not guarantee a backup in the cases where bandwidth on a single link is asymmetrical. TI-LFA only considers loop-free repair paths that are on the post-convergence path. It helps to do better capacity planning of the network.

TI-LFA algorithm allows to create a full explicit path through the network. Using fully specified path may lead to issues in larger topologies due to the number of segments along the path. Specifying the whole path is however not necessary, only a subset of the path is needed to carry the traffic to an intermediate node (release

node) which does not loop the traffic back to the protecting node. The TI-LFA algorithm constructs a SR tunnel as the repair path. TI-LFA tunnel is constructed and programmed by explicitly specifying the node (or) set of repair nodes through which the tunnel needs to traverse. The traffic is carried on the tunnel (when the primary path fails) which is also on the post convergence path.

Topology-Independent Loop Free Alternate

When the local LFA and remote LFA are enabled, there is a good coverage of the prefixes to be protected. However, for some rare topologies that do not have a PQ intersect node, both local and remote LFA will fail to find a release node to protect the failed link. Furthermore, there is no way to prefer a post-convergence path, as the two algorithms have no knowledge of the post-convergence characteristics of the LFA.

To overcome the above limitation, topology-independent LFA (TI-LFA) is supported on an SR-enabled network and provides the following support:

- **Link Protection**—The LFA provides repair path for failure of the link.
- **Local LFA**—Whenever a local LFA on the post convergence path is available, it is preferred over TI-LFA because local LFA does not require additional SID for the repair path. That is, the label for the PQ node is not needed for the release node.
- **Local LFA for extended P space**—For nodes in the extended P space, local LFA is still the most economical method for the repair path. In this case, TI-LFA is not chosen.
- **Tunnel to PQ intersect node**—This is similar to remote LFA except that the repair path is guaranteed on the post convergence path using TI-LFA.
- **Tunnel to PQ disjoint node**—This capability is unique to the TI-LFA in the case when local and remote LFA cannot find a repair path.
- **Tunnel to traverse multiple intersect or disjoint PQ nodes**—TI-LFA provides complete coverage of all prefixes, up to the platform's maximum supported labels.
- **P2P and Broadcast interfaces for the protected link**—TI-LFA protects P2P and broadcast interfaces.
- **Asymmetrical links**—The OSPF metrics between the neighbors are not the same.
- **Multi-homed (anycast) prefix protection**—The same prefix may be originated by multiple nodes and TI-LFA protects the anycast prefixes also by providing post convergence repair path.
- **Protected prefix filtering**—The route-map includes or excludes a list of prefixes to be protected and the option to limit the maximum repair distance to the release node.
- **Tiebreakers**—A subset of existing tiebreakers applicable to TI-LFA is supported.

Topology Independent Loop Free Alternate Tie-break

Local and remote LFA use default or user-configured heuristics to break the tie when there is more than one path to protect the prefix. The attributes are used to trim down the number of repair paths at the end of the TI-LFA link protection computation before the load balancing.

Local LFA and remote LFA support the following tiebreakers:

- **Linecard-disjoint**—Prefers the line card disjoint repair path.
- **Node-protecting**—Prefers node protecting repair path.
- **SRLG-disjoint**—Prefers SRLG disjoint repair path.
- **Load-sharing**—Distributes repair paths equally among links and prefixes.

When there are two repair paths for a particular prefix, the path that the output port on different line card than that of the primary port is chosen as the repair path.

- **LC-disjoint-index**—If both the repair paths are on the same line card as that of the primary path, then both paths are considered as candidates. If one of the path is on a different line card, then that path is chosen as the repair path.
- **SRLG-disjoint**—Prefers the SRLG disjoint repair path.

The SRLG ID can be configured for each interface. When there are two repair paths for a prefix, the configured SRLG ID for the repair path is compared with that of the primary path SRLG ID. If the SRLG IDs for the secondary path is different than that of the primary, that path is chosen as the repair path.

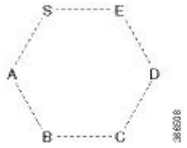
Effective with Cisco IOS-XE Release 3.18, node-protecting tie-breaker is disabled by default. Tie-breaker default and explicit tie-breaker on the same interface are mutually exclusive. The following tie-breakers are enabled by default on all LFAs:

- linecard-disjoint
- lowest-backup-metric
- SRLG-disjoint

P-Space

The set of routers that can be reached from S on the shortest path tree without traversing S-E is termed the P-space of S with respect to the link S-E.

Figure 5: A Simple Ring Topology



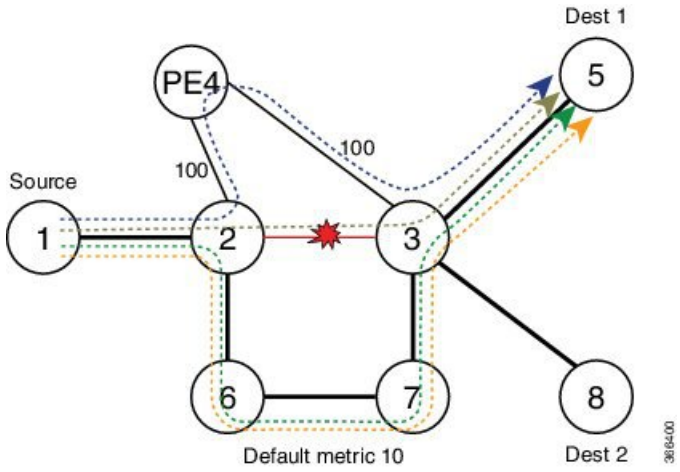
Q-Space

The set of routers from which the node E can be reached, by normal forwarding without traversing the link S-E, is termed the Q-space of E with respect to the link S-E.

Post-Convergence Path

Post convergence path is the path that OSPF uses after the link failure. TI-LFA always calculates the repair path which is the post convergence path. You can plan and dimension the post-convergence path to carry the traffic in the case of failure. TI-LFA enforces the post-convergence path by encoding it as a list of segments. The following figure shows an example of TI-LFA using post convergence path:

Figure 6: TI-LFA Using Post Convergence Path

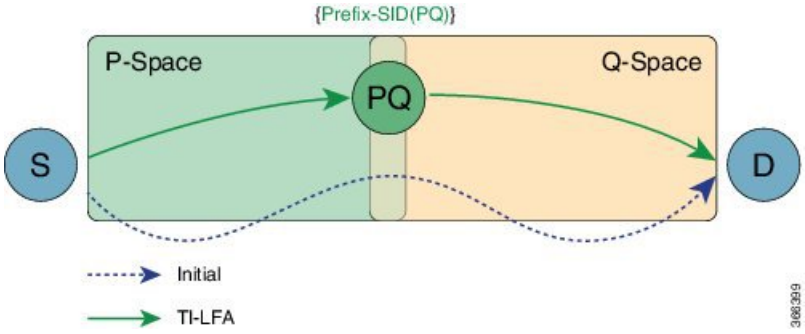


- It protects destination Node 5 on Node 2 against failure of link 2-3.
- Node 2 switches all the traffic destined to Node 5 via core links.

Per-Destination Link Protection

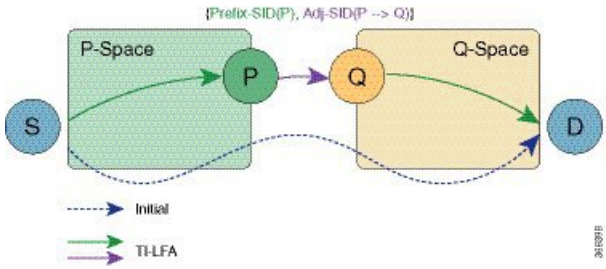
TI-LFA implementation provides per-destination link protection with the number of segments (labels)supported by the underlying hardware. The following figures show the implementation of TI-LFA:

Figure 7: TI-LFA: { Prefix-SID(PQ) }



If PQ is a direct neighbor of S, then no additional segment must be pushed.

Figure 8: TI-LFA: { Prefix-SID(P) , Adj -SID (P -> Q) }



Per Interface Loop Free Alternate Enablement

- TI-LFA can be enabled on an area basis.
- TI-LFA backup path is calculated only if TI-LFA protection is enabled on the primary interface which is to be protected. By default all the interfaces are enabled for protection.
- TI-LFA repair path is restricted by the number of labels supported by the hardware. If hardware supports only 2 labels then TI-LFA repair path can protect only those prefixes which can be protected by 2 or lesser segments. For those prefixes which need more than 2 segment remain unprotected.

Prefix Processing

Once TI-LFA path is calculated for the all the links, prefix processing starts. By default only intra and inter area prefixes are protected. For external prefixes to be protected, you need to enable segment routing globally under the OSPF level.

The primary and repair path should be of the same route type for the prefixes that are protected, that means, if the intra area needs to be protected then the TI-LFA repair path also calculates for the same intra area prefix whether the prefix is unique (or) anycast prefix.

Anycast Prefix Processing

OSPF TI-LFA also calculates the repair path for the anycast prefixes. Anycast prefixes (or) dual homed prefixes are the prefixes advertised by more than one routers. They could be intra, inter (or), external prefixes. The calculation of TI-LFA repair path for anycast prefixes is as below:

- Assume the prefix P1 is advertised by the routers R1 and R2. The prefix advertised by both the routers should be of the same route type, that is, both R1 and R2 should advertise the prefix as intra area prefix (or inter or external).
- Take the primary path is calculated towards R1 due to the lesser cost.
- When TI-LFA calculates the back up path, it calculates the post convergence path. So, post convergence path need not be towards R1. If the cost to reach R2 (in the post convergence) is shorter, then TI-LFA algorithm chooses the post convergence path towards R2. TI-LFA tunnel is formed towards R2.
- When R2 un-advertises the prefix, then the TI-LFA algorithm is re-calculated towards R1 for the repair path.

Per-Prefix Loop Free Alternate Tie-Break

IP FRR has the following tie break rules in the order given below. If you have more than one repair path available to choose the best path from, the following tie-break rules are applied. If more than one path matches all the tie break rules, then all the paths are used as repair paths.

- **Post Convergence**: Prefers backup path which is the post convergence path. This is enabled by default and user can not modify this.
- **Primary-path**: Prefers backup path from ECMP set.
- **Interface-disjoint**: Point-to-point interfaces have no alternate next hop for rerouting if the primary gateway fails. You can set the interface-disjoint attribute to prevent selection of such repair paths, thus protecting the interface.

- **Lowest-backup-metric**: Prefers backup path with lowest total metric. This is not applicable for TI-LFA since TI-LFA always chooses the back up path which is lowest cost.
- **LC-disjoint**: Prefers the back up path which is in different line card than that of the primary path.
- **Broadcast-interface-disjoint** : LFA repair paths protect links when a repair path and a protected primary path use different next-hop interfaces. However, on broadcast interfaces if the LFA repair path is computed via the same interface as the primary path and their next-hop gateways are different, in that case the node gets protected, but the link might not be. You can set the broadcast-interface-disjoint attribute to specify that the repair path never crosses the broadcast network the primary path points to, that means, it cannot use the interface and the broadcast network connected to it.
- **Load Sharing**: When more than one repair path matches the above rules, load share the backup paths. This rule also can be modified by the user.

Note The user can alter and define the tiebreak rules according to the requirement. In this way, the user can re-prioritize the sequence and/or remove some of the tie break indexes which are not needed.

Note The Lowest-backup-metric policy is not applicable for TI-LFA since TI-LFA always chooses the lowest back up path only.

You can see the above rules by using the following command:

```
R2#show ip ospf fast-reroute

            OSPF Router with ID (10.2.2.200) (Process ID 10)

Microloop avoidance is enabled for protected prefixes, delay 5000 msec

Loop-free Fast Reroute protected prefixes:

           Area        Topology name   Priority   Remote LFA Enabled     TI-LFA Enabled
              0                Base        Low                   No                Yes
    AS external                Base        Low                   No                Yes

  Repair path selection policy tiebreaks (built-in default policy):
      0  post-convergence
     10  primary-path
     20  interface-disjoint
     30  lowest-metric
     40  linecard-disjoint
     50  broadcast-interface-disjoint
    256  load-sharing

OSPF/RIB notifications:
 Topology Base: Notification Enabled, Callback Registered

Last SPF calculation started 17:25:51 ago and was running for 3 ms.
```

With the introduction of TI-LFA, the following two tie-break rules are enhanced.

- node-protection

- srlg-protection

The above two tie-break rules are not enabled by default. The user needs to configure the above mentioned tie-break policies.

Node Protection

TI-LFA node protection provides protection from node failures. Node protecting TI-LFA attempts to calculate the post conversion repair path that protects against the failure of a particular next-hop, not just the link to that particular next-hop.

Node protection is used as a tiebreaker in the implementation of the local LFA also. But when it is combined with TI-LFA, the back up path calculated post convergences with node protecting path. Per-Prefix TI-LFA node protection is disabled by default. The IPFRR TI-LFA node protection features is enabled when the corresponding tiebreak is enabled along with TI-LFA feature, that is,

```
router ospf 10
   [no] fast-reroute per-prefix ti-lfa [area <area> [disable]]
   [no] fast-reroute per-prefix tie-break node-protecting index <index>
   [no] fast-reroute per-prefix tie-break node-protecting required index <index>
```

When you enable node protection, all the other tie break rules also need to manually configured. The node protection is built over the link protection.

The difference between **node-protecting** and **node-protecting required** is in selecting the backup path. When you configure **node-protecting required**, then back up which is chosen has to be the path which does not go through the node (which is part of the link which we are protecting). If no such path is available, then no path is chosen as the backup path.

Shared Risk Link Groups Protection

A shared risk link group (SRLG) is a group of next-hop interfaces of repair and protected primary paths that have a high likelihood of failing simultaneously. The OSPFv2 Loop-Free Alternate Fast Reroute feature supports only SRLGs that are locally configured on the computing router. With the introduction of TI LFA, the post convergence path which does not share the SRLG group id with the primary path interface will be chosen. In that way, the user will be sure of the SRLG protection whenever the primary link fails.

The IPFRR TI-LFA SRLG protection features is enabled when the corresponding tiebreak is enabled along with Ti-LFA feature, that is,

```
router ospf 10
   [no] fast-reroute per-prefix ti-lfa [area <area> [disable]]
   [no] fast-reroute per-prefix tie-break srlg index <index>
   [no] fast-reroute per-prefix tie-break srlg required index <index>
```

When you enable SRLG protection, you need to manually configure all the other tie break rules. The difference between **srlg-protecting** and **srlg-protecting required** is in selecting the backup path. When you configure **srlg-protecting required**, then back up which is chosen has to be the path which does not share SRLG ID with the primary link which is protected. If no such path is available, then no path is chosen as the backup path.

Whereas, if you configure **srlg-protecting** alone then if the SRLG protection path is not available, the link protection path is chosen as the backup path. And when the SRLG protection path is available, the switchover happens to the SRLG protection path.

Node-Shared Risk Link Groups Protection

You can configure both node and SRLG protection tie breaks together. This means that the back up path needs to fulfil both the criteria of node protection as well as SRLG protection. In that case, an additional TI-LFA node-SRLG combination protection algorithm is run. The TI-LFA node-SRLG combination algorithm removes the protected node and all members of the interface with the same SRLG group when computing the post-convergence shortest path tree (SPT).

To enable node and SRLG protection tie breaks together, use the following command:

```
router ospf 10
   [no] fast-reroute per-prefix ti-lfa [area <area> [disable]]
   [no] fast-reroute per-prefix tie-break node-protecting index <index>
   [no] fast-reroute per-prefix tie-break srlg index <index>
```

The following show command is used to display the tie break policy:

```
R3#show ip ospf fast-reroute

            OSPF Router with ID (10.3.3.33) (Process ID 10)

Loop-free Fast Reroute protected prefixes:

           Area        Topology name   Priority   Remote LFA Enabled     TI-LFA Enabled
              0                Base        Low                   No                 No
              1                Base        Low                   No                 No
           1000                Base        Low                   No                 No
    AS external                Base        Low                   No                 No

  Repair path selection policy tiebreaks:
      0  post-convergence
     60  node-protecting
     70  srlg
    256  load-sharing

OSPF/RIB notifications:
 Topology Base: Notification Disabled, Callback Not Registered

Last SPF calculation started 00:00:06 ago and was running for 2 ms.
```

How to Configure Topology Independent Loop Free Alternate Fast Reroute

Enabling Topology Independent Loop Free Alternate Fast Reroute

By default, TI-LFA is disabled. You can use protocol enablement to enable TI-LFA.

Protocol enablement: Enables TI-LFA in router OSPF mode for all the OSPF areas. Perform the following steps to enable TI-LFA FRR.

```
[no] fast-reroute per-prefix ti-lfa [ area <area> disable]
```

```
  router ospf <process>
  fast-reroute per-prefix enable area <area> prefix-priority {low | high}
  fast-reroute per-prefix ti-lfa [ area <area> disable]
```

You can also use interface command to enable or disable IP FRR on specific interfaces.

```
interface <interface>
ip ospf fast-reroute per-prefix protection disable
ip ospf fast-reroute per-prefix candidate disable
ip ospf fast-reroute per-prefix protection ti-lfa [disable]
```



Note

- When TI-LFA is configured on the OSPF router and area wide, area specific configuration takes precedence.
- To protect external prefixes, TI-LFA should be enabled globally.

Configuring Topology Independent Loop Free Alternate Fast Reroute

This task describes how to enable per-prefix Topology Independent Loop-Free Alternate (TI-LFA) computation to converge traffic flows around link, node, and SRLG failures. TI-LFA can be configured on instance or area level inherited by lower levels. You can enable or disable per prefix FRR per interface level which is applicable for TI-LFA also.

Before you begin to configure, ensure that the following topology requirements are met:

- Router interfaces are configured as per the topology.
- Routers are configured with OSPF.
- Segment routing is enabled globally as well as under OSPF level.

1. Enables OSPF routing for the specified routing process and enters in router configuration mode.

   ```
   Device(config)# router ospf 10
   ```

2. Enables FRR.

   ```
   Device(config-router)# fast-reroute per-prefix enable prefix-priority low
   ```

3. Enables TI-LFA.

   ```
   Device(config-router)# fast-reroute per-prefix ti-lfa
   ```

4. Enables TI-LFA on the specific area.

   ```
   Device(config-router)# fast-reroute per-prefix ti-lfa area 0
   ```

5. Exits the TI-LFA mode.

   ```
   Device(config-router)# exit
   ```

6. Enters the interface mode.

   ```
   Device(config)#interface ethernet 0/0
   ```

7. If you do not wish to enable FRR on a specific inteface, use the protection disable command.

   ```
   Device(config-if)#ip ospf fast-reroute per-prefix protection disable
   ```

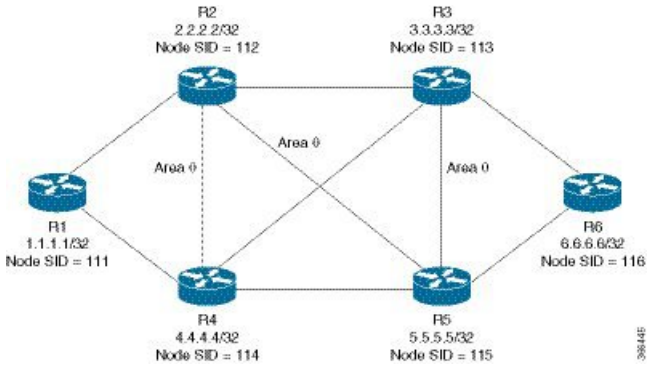
8. If you do not wish a specific interface to be enabled as a repair path, use the candidate disable command.

   ```
   Device(config-if)#ip ospf fast-reroute per-prefix candidate disable
   ```

Configuring Topology Independent Fast Reroute Tie-breaker

You need to enable segment routing on all the routers with prefix SIDs configured for all the nodes. Use the following topology as a reference to understand the configuration.

Figure 9: Configuration Example



Let us take the device R2 which is protecting the link between R2 and R3. The configuration at R2:

```
router ospf 10
fast-reroute per-prefix enable prefix-priority low
fast-reroute per-prefix ti-lfa
segment-routing mpls
segment-routing area 0 mpls
fast-reroute per-prefix enable prefix-priority low
fast-reroute per-prefix ti-lfa
fast-reroute per-prefix ti-lfa area 0
fast-reroute per-prefix tie-break node-protecting index 60
fast-reroute per-prefix tie-break srlg index 70
mpls traffic-eng router-id Loopback1
mpls traffic-eng area 0

interface GigabitEthernet4   //interface connecting to the router 4
ip address 10.101.4.4 255.255.255.0
ip ospf 10 area 0
ip ospf network point-to-point
srlg gid 10
negotiation auto

interface GigabitEthernet3  //interface connecting to the router 3
ip address 10.101.3.3 255.255.255.0
ip ospf 10 area 0
ip ospf network point-to-point
srlg gid 10
negotiation auto

interface GigabitEthernet5   //interface connecting to the router 2
ip address 10.101.5.5 255.255.255.0
ip ospf 10 area 0
ip ospf network point-to-point
srlg gid 20
negotiation auto



interface loopback2
```

```
ip address 10.2.2.2/32
ip ospf 10 area 0
```



Note In all the other devices, configuration of segment routing and assignment of connected prefix SIDs need to be done.

How Node Protection Works: Using the same topology as an example, let us take the case where you are protecting the link between R2 and R3 and also the prefix which is leant from R6. In that case, let us assume that the primary path for the prefix is via R2-R3. So, our primary path is R2---R3---R6 and we are protecting the link R2---R3.

In this scenario, only link-protection is configured and enabled. When you enable TI-LFA under OSPF process, then you get the following paths provided the cost for all the paths are equal:

R2----R4----R5---R6

R2---R5----R3---R6

R2----R5---R6

If you have only link protection configured, then all the three paths will be chosen and they will share the load amongst them.

If you wish to configure node protection, then the backup would be calculated in such a way that the back up path does not contain the node that you are protecting. In this example, the node R3 in the back up is not required. As a result, only the following two paths would be chosen as the back up paths:

R2----R4----R5---R6

R2----R5---R6

It is possible that R2---R5---R3---R6 have the lesser cost than the above two paths. But since the node protection is configured, only the paths amongst the above two will be considered.

How SRLG Protection Works: SRLG protection further eliminates the back up paths in a such a way that the primary path and the backup does not share the same SRLG ID. Suppose the following back up paths are available:

R2----R4----R5---R6

R2----R5---R6

Then, the SRLG ID of (R2----R4) and (R2----R5) are compared against the primary interface (R2----R3) which is 10. It is noticed that only the interface R2----R5 has different SRLG ID which is 20. So, only the backup path R2---R5---R6 will be chosen.

Verifying Topology Independent Fast Reroute Tunnels

You can use the following command, to check the TI LFA tunnels:

```
Device#show ip ospf fast-reroute ti-lfa tunnels


            OSPF Router with ID (10.2.2.200) (Process ID 10)

                        Area with ID (0)
```

```
                    Base Topology (MTID 0)

Tunnel                Interface         Next Hop         Mid/End Point    Label
--------------------------------------------------------------------------------
MPLS-SR-Tunnel2       Et1/1             10.7.0.7         10.1.1.1         16020
MPLS-SR-Tunnel6       Et0/3             10.8.0.0         10.3.3.3         16003
MPLS-SR-Tunnel7       Et1/1             10.7.0.7         10.1.1.1         16020
                                                         10.5.5.5         16005
                                                         10.3.3.3         16003
MPLS-SR-Tunnel5       Et0/3             10.8.0.0         10.5.5.5         16005
MPLS-SR-Tunnel1       Et1/1             10.7.0.7         10.1.1.1         16020
                                                         10.5.5.5         16005
MPLS-SR-Tunnel3       Et1/1             10.7.0.7         10.6.6.6         16006
```

You can use the following command, to check the route in OSPF routing table with primary and repair path:

```
Device#show ip ospf rib 10.6.6.6

            OSPF Router with ID (10.2.2.200) (Process ID 10)

                Base Topology (MTID 0)

OSPF local RIB
Codes: * - Best, > - Installed in global RIB
LSA: type/LSID/originator

*>  10.6.6.6/32, Intra, cost 31, area 0
     SPF Instance 19, age 02:12:11
      contributing LSA: 10/10.0.0.0/10.6.6.6 (area 0)
     SID: 6
     CSTR Local label: 0
     Properties: Sid, LblRegd, SidIndex, N-Flag, TeAnn
     Flags: RIB, HiPrio
      via 10.7.0.7, Ethernet1/1 label 16006
       Flags: RIB
       LSA: 1/10.6.6.6/10.6.6.6
      PostConvrg repair path via 10.3.3.3, MPLS-SR-Tunnel6 label 16006, cost 81, Lbl cnt 1
       Flags: RIB, Repair, PostConvrg, IntfDj, LC Dj
       LSA: 1/10.6.6.6/10.6.6.6
```

You can use the following command, to display the route in the IP routing table:

```
Device#show ip route 10.6.6.6
Routing entry for 10.6.6.6/32
  Known via "ospf 10", distance 110, metric 31, type intra area
  Last update from 10.7.0.7 on Ethernet1/1, 00:25:14 ago
 SR Incoming Label: 16006
  Routing Descriptor Blocks:
  * 10.7.0.7, from 10.6.6.6, 00:25:14 ago, via Ethernet1/1, merge-labels
      Route metric is 31, traffic share count is 1
      MPLS label: 16006
      MPLS Flags: NSF
      Repair Path: 10.3.3.3, via MPLS-SR-Tunnel6
```

Debugging Topology Independent Loop Free Alternate Fast Reroute

You can use the following commands to debug TI-LFA FRR:

```
debug ip ospf fast-reroute spf
debug ip ospf fast-reroute spf detail
debug ip ospf fast-reroute rib
debug ip ospf fast-reroute rib [<access-list>]
```

Examples: OSPFv2 Link-Protection Topology Independent Loop Free Alternate Fast Reroute

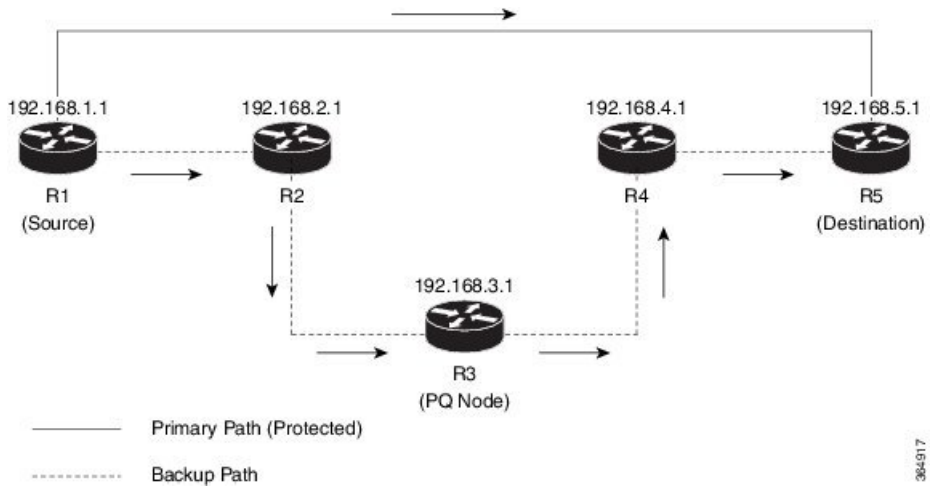
The following are the examples for the OSPFv2 Link-Protection TI-LFA FRR.

Example: Configuring Topology Independent Loop Free Alternate Fast Reroute

This example shows how to configure TI-LFA for segment routing TE tunnels using single or disjoint PQ nodes. The following are the two topologies used:

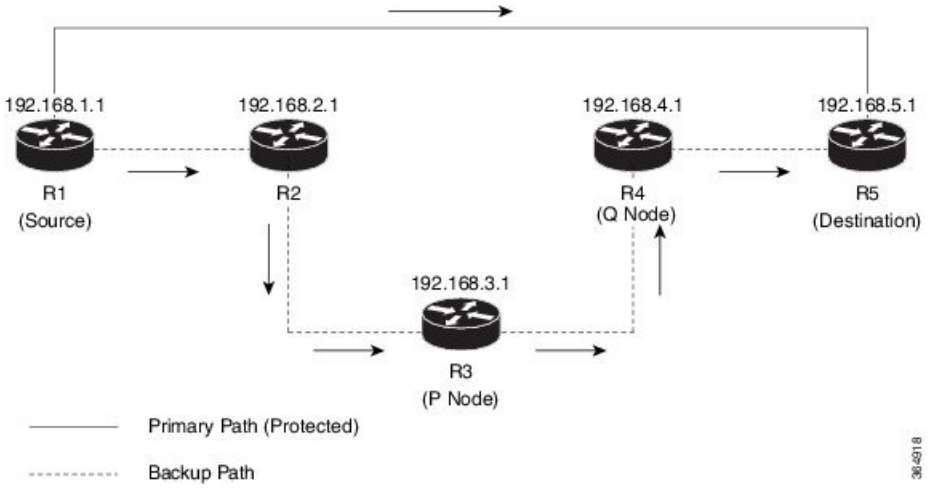
- Topology 1: A single PQ Node and therefore has two SIDs from the source router, R1 through the PQ Node to the destination router, R5.

Figure 10: Topology 1: Single PQ Node



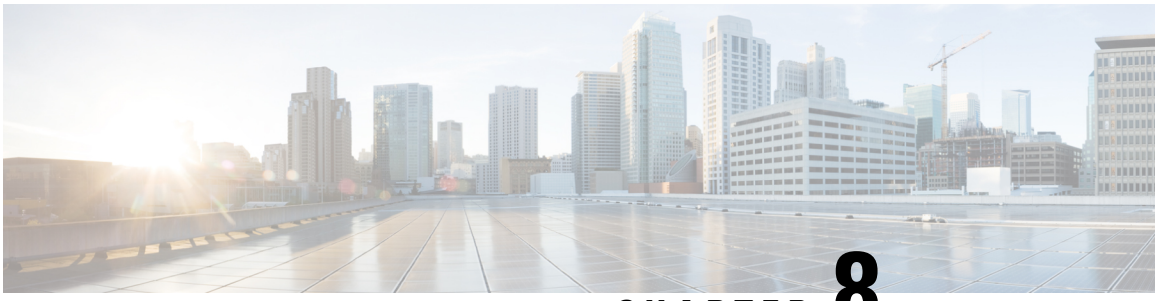
- Topology 2: Disjoint PQ Nodes and therefore consists of three SIDs from the source router R1, through the P Node and the Q Node to the destination router, R5.

Figure 11: Topology 2: Disjoint PQ Nodes



Configure TI-LFA for OSPF on the source router (R1) interface connecting to the destination router (R5).

```
Device(config)# router ospf 10
Device(config-router)# fast-reroute per-prefix enable prefix-priority low
Device(config-router)# fast-reroute per-prefix ti-lfa
Device(config-router)# fast-reroute per-prefix ti-lfa area 0
Device(config-router)# exit
```



CHAPTER 8

Segment Routing Traffic Engineering With OSPF

This chapter describes how Segment Routing traffic engineering can be implemented using OSPF.

- Feature Information for Segment Routing Traffic Engineering With OSPF, on page 85
- Restrictions for Segment Routing Traffic Engineering With OSPF, on page 86
- Information About Segment Routing Traffic Engineering With OSPF, on page 86
- How to Configure Segment Routing Traffic Engineering With OSPF, on page 94
- Verifying Configuration of the SR-TE Tunnels, on page 102

Feature Information for Segment Routing Traffic Engineering With OSPF

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 6: Feature Information for Segment Routing Traffic Engineering With OSPF

Feature Name	Releases	Feature Information
Segment Routing Traffic Engineering With OSPF	Cisco IOS XE Amsterdam 17.3.2	A Traffic Engineered (TE) tunnel is a container of TE LSP(s) instantiated between the tunnel ingress and the tunnel destination. A TE tunnel may instantiate one or more SR-TE LSP(s) that are associated with the same tunnel. The following commands were added or modified: **show mpls traffic-eng tunnels**, **tunnel mpls traffic-eng path-option 10 dynamic segment-routing**, **tunnel mpls traffic-eng path-option 10 segment-routing**, **tunnel mpls traffic-eng path-option 10 explicit name MIXED_PATH segment-routingtunnel mpls traffic-eng path-option 10 explicit name IP_PATH1 segment-routingtunnel mpls traffic-eng path-option 10 explicit name NODE_PATH segment-routing**.

Restrictions for Segment Routing Traffic Engineering With OSPF

- Segment Routing Traffic Engineering is supported only on OSPFv2.
- SR-TE is not supported on broadcast interfaces; it is supported only point-to-point interfaces.
- Only one instance of protocol should be enabled for TE at a given point of time.

Information About Segment Routing Traffic Engineering With OSPF

A Traffic Engineered (TE) tunnel is a container of TE LSP(s) instantiated between the tunnel ingress and the tunnel destination. A TE tunnel may instantiate one or more SR-TE LSP(s) that are associated with the same tunnel. The SR-TE LSP path may not necessarily follow the same IGP path to a destination node. In this case, the SR-TE path can be specified a set of prefix-SID(s) and/or adjacency-SID(s) of nodes and/or links to be traversed by the SR-TE LSP.

The head-end imposes the corresponding MPLS label stack on to outgoing packets to be carried over the tunnel. Each transit node along the SR-TE LSP path uses the incoming top label to select the next-hop, pop or swap the label, and forward the packet to the next node with the remainder of the label stack, until the packet reaches the ultimate destination. OSPF provides TE with the topology and SR related information. SR related information include SRGB/prefix/Adjacency SIDs of all nodes/links with SR enabled in the network.

Benefits of Using Segment Routing Traffic Engineering With OSPF

Segment routing traffic engineering offers a comprehensive support for all useful optimizations and constraints, for example:

- Latency
- Bandwidth
- Disjointness
- Resource avoidance

OSPFv2 provides the following functionalities for SR-TE:

- OSPFv2 provides SR information along with TE topology information to TE module.
- TE uses this information to construct SR TE path/tunnel comprising of one or more segments - with the combination of prefix and/or adjacency segments.
- For the prefixes TE is interested in, OSPF provides first hop resolution to setup the forwarding plane.
- SR TE tunnels are also advertised back into OSPF (like RSVP TE tunnels) for diverting traffic over the SR-TE tunnels.

OSPFv2 Segment Routing Traffic Engineering Functionalities

OSPFv2 perform the following functionalities for SR-TE:

- OSPFv2 provides SR information along with TE topology information to TE module.
- TE uses this information to construct SR TE path/tunnel comprising of one or more segments - with the combination of prefix and/or adjacency segments.
- For the prefixes TE is interested in, OSPF provides first hop resolution to setup the forwarding plane.
- SR TE tunnels are also advertised back into OSPF (like RSVP TE tunnels) for diverting traffic over the SR-TE tunnels.

Protected Adjacency SID

Segment routing creates protected adjacency SID for point to point to point interfaces and broadcast interfaces. It advertises them to the extended link-state advertisement (LSA) along with the unprotected adjacency SID. Protected adjacency SID can have a repair path, but it is not guaranteed to have a repair path.

Traffic Engineering Interfaces

In order to support SR-TE functionality, TE interfaces with various components, and with IGP (OSPF and ISIS) to distribute and receive information on TE topology. For SR-TE support, OSPF needs to additionally provide SR information to TE that it had received through various LSAs, for example,

- Router Information LSA
- Extended Prefix LSA
- Extended Link LSA

TE interfaces distribute information, such as bandwidth resources, constraints, capabilities, and other attributes, associated with the links that are configured for TE. The link information is distributed to other routers using opaque LSAs and is used by TE to create a local topology database. The topology database is a key element in allowing TE to compute a suitable constraint-based path for establishing an LSP. TE also interfaces with the IGP to notify when a TE headend interface can be considered for routing packets.

Unnumbered Support

IS-IS description of an unnumbered link does not contain remote interface ID information. The remote interface ID of an unnumbered link is required to include the unnumbered link as part of the SR-TE tunnel.

Segment Routing Traffic Engineering Support for Forwarding Adjacency

MPLS TE forwarding adjacency feature is supported by OSPF. In this, TE tunnel is considered as a link in the IGP network. TE tunnel interfaces are advertised in the IGP network like any other links. Routers can then use these links to compute the shortest path tree (SPT).

Note This feature is not supported with the SR-TE tunnels.

Segment Routing Traffic Engineering Support for Auto-route Announce

MPLS TE auto-route announce feature is supported by OSPF, that uses TE Tunnel as the first-hop, if the node is reachable via that tunnel. It allows the traffic to the nodes that are downstream to the tail-end of the TE tunnel flows through the tunnel. OSPF supports auto-route over the SR-TE tunnels similar to the MPLS TE tunnels setup using RSVP.

The TE tunnel that instantiates an SR-TE LSP can be Auto-route Announced (AA) into IGP (OSPF and ISIS) as an IGP shortcut. The IGP uses the TE tunnel as next hop and installs routes in RIB for all IP prefixes whose shortest path falls behind the TE tunnel destination. Auto-route announce for of TE tunnels is supported to carry IPV4 prefixes.

Auto-route Announce IP2MPLS

The auto-routeIP2MPLS feature is introduced for SR tunnels to avoid potential packet from looping indefinitely between the SR-TE tunnel headend/ingress and a node that is pointing/routing the packet back to the headend/ingress.

The solution consists in the headend programming in forwarding two sets of path(s) for the prefixes that are mapped over the SR-TE tunnel. The first is the pure IP route for the prefix(es) mapped on the and having the outgoing interface as the tunnel interface. This allows mapping IP traffic directly over the tunnel. The second is the MPLS path for the prefixes mapped on the tunnel. For this the prefix-SID label is programmed with the IGP shortest path outgoing interface(s), that is, non tunnel output interfaces.

SR-TE LSP Instantiation

A Traffic Engineered (TE) tunnel is a container of one or more instantiated TE LSPs. An SR-TE LSP is instantiated by configuring ‘segment-routing’ on the path-option of the TE tunnel. The traffic mapped to the tunnel is forwarded over the primary SR-TE instantiated LSP.

Multiple path-options can also be configured under the same tunnel. Each path-option is assigned a preference index or a path-option index that is used to determine the more favorable path-option for instantiating the primary LSP—the lower the path-option preference index, the more favorable the path-option. The other less favorable path-options under the same TE tunnel are considered secondary path-options and may be used once the currently used path-option is invalidated (for example, due to a failure on the path.

Note A forwarding state is maintained for the primary LSP only.

Tunnel Path Affinity Validation

The affinity of a tunnel path can be specified using the command **tunnel mpls traffic-eng affinity** under the tunnel interface.

The head-end validates that the specified SR path is compliant with the configured affinity. This necessitates that the paths of each segment of the SR path be validated against the specified constraint. The path is declared invalid against the configured affinity constraints if at least a single segment of the path does not satisfy the configured affinity.

SR-TE Traffic Load Balancing

SR-TE tunnels support the following load-balancing options:

Load Balancing on Port Channel TE Links

Port Channel interfaces carry the SR-TE LSP traffic. This traffic load balances over port channel member links as well as over bundle interfaces on the head or mid of an SR-TE LSP.

Load Balancing on Single Tunnel

While using the equal cost multi path protocol (ECMP), the path to a specific prefix-SID may point to multiple next-hops. And if the SR-TE LSP path traverses one or more prefix-SIDs that have ECMP, the SR-TE LSP traffic load-balances on the ECMP paths of each traversed prefix-SID from the head-end or any midpoint traversed node along the SR-TE LSP path.

Load Balancing on Multiple Tunnels

Multiple TE tunnels can be used as next-hop paths for routes to specific IP prefixes either by configuring static route on multiple tunnels, or auto-route announcing multiple parallel tunnels to the same destination. In such cases, the tunnels share the traffic load equally or load balance traffic on multiple parallel tunnels. It is also possible to allow Unequal Load Balance (UELB) with an explicit per tunnel configuration at the tunnel head-end. In this case, the tunnel load-share is passed from MPLS-TE to forwarding plane.

The tunnel load-share feature continues to work for TE tunnels that instantiate the SR-TE LSPs.

SR-TE Tunnel Reoptimization

TE tunnel reoptimization occurs when the head-end determines that there is a more optimal path available than the one currently used. For example, in case of a failure along the SR-TE LSP path, the head-end could detect and revert to a more optimal path by triggering reoptimization.

Tunnels that instantiate SR-TE LSP can re-optimize without affecting the traffic carried over the tunnel.

Re-optimization can occur because:

- The explicit path hops used by the primary SR-TE LSP explicit path are modified.
- The head-end determines the currently used path-option are invalid due to either a topology path disconnect, or a missing SID in the SID database that is specified in the explicit-path.
- A more favorable path-option (lower index) becomes available.

When the head-end detects a failure on a protected SR adjacency-SID that is traversed by an SR-TE LSP, it starts the invalidation timer. If the timer expires and the head-end is still using the failed path because it is unable to reroute on a different path, the tunnel state is brought ‘down’ to avoid a null route from being sent along with the traffic. Once the tunnel is down, services on the tunnel converge to take a different path.

The following is a sample output of a manual reoptimization example. In this example, the path-option is changed from **10** to **20**.

```
Router# mpls traffic-eng reoptimize tunnel 1 path-option 20
The targeted path-option is not in lock down mode. Continue? [no]: yes
Router# show mpls traffic-eng tunnels tunnel1
Name: R1_t1                           (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 20, (SEGMENT-ROUTING) type explicit IP_PATH (Basis for Setup)
    path option 10, (SEGMENT-ROUTING) type dynamic
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 20 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours, 9 minutes
      Time since path change: 14 seconds
      Number of LSP IDs (Tun_Instances) used: 1819
    Current LSP: [ID: 1819]
      Uptime: 17 seconds
      Selection: reoptimization
    Prior LSP: [ID: 1818]
      ID: path option unknown
      Removal Trigger: reoptimization completed
  Tun_Instance: 1819
  Segment-Routing Path Info (isis level-1)
    Segment0[Node]: 10.4.4.4, Label: 114
    Segment1[Node]: 10.5.5.5, Label: 115
    Segment2[Node]: 10.6.6.6, Label: 116
```

SR-TE with Lockdown Option

The **lockdown** option prevents SR-TE from re-optimizing to a better path. However, it does not prevent signaling the existence of a new path.

```
interface Tunnel1
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng path-option 10 segment-routing lockdown
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10                        (Tunnel1) Destination:
10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (LOCKDOWN) type segment-routing (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
```

```
  Path Selection:
   Protection: any (default)
  Path-invalidation timeout: 45000 msec (default), Action: Tear
  AutoRoute: enabled  LockDown: enabled  Loadshare: 10 [200000000]
  auto-bw: disabled
  Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
Active Path Option Parameters:
  State: segment-routing path option 10 is active
  BandwidthOverride: disabled  LockDown: enabled   Verbatim: disabled
History:
  Tunnel:
    Time since created: 6 days, 19 hours, 22 minutes
    Time since path change: 1 minutes, 26 seconds
    Number of LSP IDs (Tun_Instances) used: 1822
  Current LSP: [ID: 1822]
    Uptime: 1 minutes, 26 seconds
    Selection: reoptimization
  Prior LSP: [ID: 1821]
    ID: path option unknown
    Removal Trigger: configuration changed
Tun_Instance: 1822
Segment-Routing Path Info (isis  level-1)
  Segment0[Node]: 10.6.6.6, Label: 116
```

SR-TE Tunnel Protection

Protection for SR TE tunnels can take any of the following alternatives:

IP-FRR Local Repair Protection

On an SR-TE LSP head-end or mid-point node, IP-FRR is used to compute and program the backup protection path for the prefix-SID or adjacency-SID label.

With IP-FRR, backup repair paths are pre-computed and pre-programmed by IGPs *before* a link or node failure. The failure of a link triggers its immediate withdrawal from the TE topology (link advertisement withdrawal). This allows the head-end to detect the failure of an SR-TE LSP traversing the failed adjacency-SID.

When a protected adjacency-SID fails, the failed adjacency-SID label and associated forwarding are kept functional for a specified period of time (5 to 15 minutes) to allow all SR TE tunnel head-ends to detect and react to the failure. Traffic using the adjacency-SID label continues to be FRR protected even if there are subsequent topology updates that change the backup repair path. In this case, the IGPs update the backup repair path while FRR is active to reroute traffic on the newly-computed backup path.

When the primary path of a protected prefix-SID fails, the PLR reroutes to the backup path. The head-end remains transparent to the failure and continues to use the SR-TE LSP as a valid path.

IP-FRR provides protection for adjacency and prefix-SIDs against link failures only.

Tunnel Path Protection

Path protection is the instantiation of one or more standby LSPs to protect against the failure of the primary LSP of a single TE tunnel.

Path protection protects against failures by pre-computing and pre-provisioning secondary paths that are failure diverse with the primary path-option under the same tunnel. This protection is achieved by computing a path that excludes prefix-SIDs and adjacency-SIDs traversed by the primary LSP or by computing a path that excludes SRLGs of the primary SR-TE LSP path.

In the event of a failure of the primary SR-TE LSP, at least one standby SR-TE LSP is used for the tunnel. Multiple secondary path-options can be configured to be used as standby SR-TE LSPs paths.

SR-TE LSP Path Verification

SR-TE tunnel functionality requires that the head-end perform initial verification of the tunnel path as well as the subsequent tracking of the reachability of the tunnel tail-end and traversed segments.

Path verification for SR-TE LSP paths is triggered whenever MPLS-TE is notified of any topology changes or SR SID updates.

The SR-TE LSP validation steps consist of the following checks:

Topology Path Validation

The head-end validates the path of an SR-TE LSP for connectivity against the TE topology. MPLS-TE head-end checks if links corresponding to the adjacency SIDs are connected in the TE topology.

For newly-instantiated SR-TE LSPs, if the head-end detects a discontinuity on any link of the SR-TE path, that path is considered invalid and is not used. If the tunnel has other path-options with valid paths, those paths are used to instantiate the tunnel LSP.

For TE tunnels with existing instantiated SR-TE LSP, if the head-end detects a discontinuity on any link, the head-end assumes a fault has occurred on that link. In this case, the local repair protection, such as the IP FRR, come in to effect. The IGPs continue to sustain the protected adjacency label and associated forwarding after the adjacency is lost for some time. This allows the head-ends enough time to reroute the tunnels onto different paths that are not affected by the same failure. The head-end starts a tunnel invalidation timer once it detects the link failure to attempt to reroute the tunnel onto other available path-options with valid paths.

If the TE tunnel is configured with other path-options that are not affected by the failure and are validated, the head-end uses one of those path-options to reroute (and re-optimize) the tunnel by instantiating a new primary LSP for the tunnel using the unaffected path.

If no other valid path-options exist under the same tunnel, or if the TE tunnel is configured with only one path-option that is affected by the failure, the head-end starts an invalidation timer after which it brings the tunnel state to ‘down’. This action avoids a null route from being sent along with traffic flowing over the affected SR-TE LSP, and allows services riding over the tunnel to reroute over different available paths at the head-end. There is an invalidation drop configuration that keeps the tunnel 'up', but drops the traffic when the invalidation timer expires.

For intra-area SR-TE LSPs, the head-end has full visibility over the LSP path, and validates the path to the ultimate LSP destination. However, for inter-area LSPs, the head-end has partial visibility over the LSP path—only up to the first ABR. In this case, the head-end can only validate the path from the ingress to the first ABR. Any failure along the LSP beyond the first ABR node is invisible to the head-end, and other mechanisms to detect such failures, such as BFD over LSP are assumed.

SR SID Validation

SID hops of an SR-TE LSP are used to determine the outgoing MPLS label stack to be imposed on the outgoing packets carried over the SR-TE LSP of a TE tunnel. A database of global and local adjacency-SIDs is populated from the information received from IGPs and maintained in MPLS-TE. Using a SID that is not available in the MPLS TE database invalidates the path-option using the explicit-path. The path-option, in this case, is not used to instantiate the SR TE LSP. Also, withdrawing, adding, or modifying a SID in the MPLS-TE SID-database, results in the MPLS-TE head-end verifying all tunnels with SR path-options (in-use or secondary) and invokes proper handling.

LSP Egress Interface

When the SR-TE LSP uses an adjacency-SID for the first path hop, TE monitors the interface state and IGP adjacency state associated with the adjacency-SID and the node that the SR-TE LSP egresses on. If the interface or adjacency goes down, TE can assume a fault occurred on the SR-TE LSP path and take the same reactive actions described in the previous sections.



Note When the SR-TE LSP uses a prefix-SID for the first hop, TE cannot directly infer on which interface the tunnel egresses. TE relies on the IP reachability information of the prefix to determine if connectivity to the first hop is maintained.

IP Reachability Validation

MPLS-TE validates that the nodes corresponding to the prefix-SIDs are IP reachable before declaring the SR path valid. MPLS-TE detects path changes for the IP prefixes corresponding to the adjacency or prefix SIDs of the SR-TE LSP path. If the node announcing a specific SID loses IP reachability. due to a link or node failure, MPLS-TE is notified of the path change (no path). MPLS-TE reacts by invalidating the current SR-TE LSP path, and may use other path-options with a valid path, if any to instantiate a new SR-TE LSP.



Note Since IP-FRR does not offer protection against failure of a node that is being traversed by an SR-TE LSP (such as, a prefix-SID failure along the SR-TE LSP path), the head-end immediately reacts to IP route reachability loss for prefix-SID node by setting the tunnel state to ‘down’ and removes the tunnel forwarding entry if there are no other path-options with valid path for the affected tunnel.

Tunnel Path Resource Avoidance Validation

You can specify a set of addresses to be validated as excluded from being traversed by SR-TE tunnel packets. To achieve this, the head-end runs the per-segment verification checks and validates that the specified node, prefix or link addresses are indeed excluded from the tunnel in the SR path. The tunnel resource avoidance checks can be enabled per path using the commands below. The list of addresses to be excluded are defined and the name of the list is referenced in the path-option.

```
interface tunnel100
 tunnel mpls traffic-eng path-option 1 explicit name EXCLUDE segment-routing
ip explicit-path name EXCLUDE enable
 exclude-address 192.168.0.2
 exclude-address 192.168.0.4
 exclude-address 192.168.0.3
!
```

SR-TE LSP Explicit Null

MPLS-TE tunnel head-end does not impose explicit-null at the bottom of the stack. When penultimate hop popping (PHP) is enabled for SR prefix SIDs or when an adjacency SID is the last hop of the SR-TE LSP, the packet may arrive at the tail-end without a transport label. However, in some cases, it is desirable that the packet arrive at the tail-end with explicit-null label, and in such case, the head-end will impose an explicit-null label at the top of the label stack.

Verbatim Path Support

MPLS TE LSPs usually require that all the nodes in the network are TE aware which means that they have IGP extensions to TE in place. However, some network administrators want the ability to build TE LSPs to traverse nodes that do not support IGP extensions to TE, but that do support RSVP extensions to TE. Verbatim LSPs are helpful when all or some of the intermediate nodes in a network do not support IGP extensions for TE.

When this feature is enabled, the IP explicit path is not checked against the TE topology database. Since the TE topology database is not verified, a Path message with IP explicit path information is routed using the shortest path first (SPF) algorithm for IP routing.

How to Configure Segment Routing Traffic Engineering With OSPF

Perform the following steps to configure Segment Routing Traffic Engineering With OSPF.

Enabling Segment Routing Traffic Engineering With OSPF

OSPF Segment Routing traffic engineering is enabled when the segment-routing is enabled along with mpls traffic engineering. SR-TE support is turned on in an area when you enable SR & MPLS TE in that area.

```
router ospf 10
 router-id 10.10.10.2
 segment-routing mpls
  mpls traffic-eng area 0
```

Configuring the Path Option for a TE Tunnel

When the path-option type for an operational SR tunnel is changed from SR to non-SR (for example, **dynamic**), the existing forwarding entry of the tunnel is deleted.

Segment Routing can be enabled or disabled on an existing secondary or an in-use path-option. If the tunnel uses a signaled RSVP-TE explicit path-option and segment routing is enabled on that tunnel, the RSVP-TE LSP is torn, and the SR-TE LSP is instantiated using the same path-option. Conversely, if segment routing is disabled on a path-option that is in use by the primary LSP, the tunnel goes down intermittently and a new RSVP-TE LSP will be signaled using the same explicit path.

If the segment-routing path-option is enabled on a secondary path-option (that is, not in-use by the tunnel's primary LSP), the tunnel is checked to evaluate if the newly specified SR-TE LSP path-option is valid and more favorable to use for the tunnel primary LSP.

```
Device(config)# interface tunnel 100
Device(config-if)# tunnel mpls traffic-eng path-option 1 explicit name foo segment-routing
Device(config-if)# tunnel mpls traffic-eng path-option 2 dynamic segment-routing
Device(config-if)# tunnel mpls traffic-eng path-option 3 segment-routing
```

Configuring SR Explicit Path Hops

The following explicit path hops are supported in SR-TE:

- IP addresses
- MPLS labels
- Mix of IP addresses and MPLS labels

For intra-area LSPs, the explicit path can be specified as a list of IP addresses:

```
Device(config)# ip explicit-path name foo
Device(config-ip-expl-path)# index 10 next-address 10.1.1.1 node address
Device(config-ip-expl-path)# index 20 next-address 10.12.12.2 link address
```



Note When using IP unnumbered interfaces, you cannot specify next hop address as an explicit path index. It should be node address or label.

The explicit path can also be specified as segment-routing SIDs:

```
Device(config)# ip explicit-path name foo
Device(config-ip-expl-path)# index 10 next-label 20
```

Configuring Tunnel Path Affinity Validation

The affinity of a tunnel path can be specified using the command **tunnel mpls traffic-eng affinity** under the tunnel interface.

The head-end validates that the specified SR path is compliant with the configured affinity. This necessitates that the paths of each segment of the SR path be validated against the specified constraint. The path is declared invalid against the configured affinity constraints if at least a single segment of the path does not satisfy the configured affinity.

```
interface Tunnel1
 no ip address
 tunnel mode mpls traffic-eng
 tunnel destination 10.5.5.5
 tunnel mpls traffic-eng priority 5 5
 tunnel mpls traffic-eng bandwidth 100
 tunnel mpls traffic-eng affinity 0x1 mask 0xFFFF
        tunnel mpls traffic-eng path-option 10 dynamic segment-routing
Router# show tunnel ??
Name: R1_t1                             (Tunnel1) Destination: 10.5.5.5
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type dynamic (Basis for Setup, path weight 20)
  Config Parameters:
    Bandwidth: 100      kbps (Global)  Priority: 5  5   Affinity: 0x1/0xFFFF
    Metric Type: TE (default)
    Path Selection:
     Protection: any (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
    Hop Limit: disabled
    Cost Limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear
    AutoRoute: disabled LockDown: disabled Loadshare: 100 [0] bw-based
    auto-bw: disabled
```

```
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: dynamic path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  Node Hop Count: 2
  History:
    Tunnel:
      Time since created: 10 minutes, 54 seconds
      Time since path change: 34 seconds
      Number of LSP IDs (Tun_Instances) used: 55
    Current LSP: [ID: 55]
      Uptime: 34 seconds
    Prior LSP: [ID: 49]
      ID: path option unknown
      Removal Trigger: tunnel shutdown
  Tun_Instance: 55
  Segment-Routing Path Info (isis  level-1)
    Segment0[Link]: 192.168.2.1 - 192.168.2.2, Label: 46
    Segment1[Link]: 192.168.4.2 - 192.168.4.1, Label: 49
```

Configuring Affinity on an Interface

Perform the following steps to configure affinity on an interface:

```
interface GigabitEthernet2
 ip address 192.168.2.1 255.255.255.0
 ip router isis 1
 negotiation auto
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 isis network point-to-point
 ip rsvp bandwidth
```

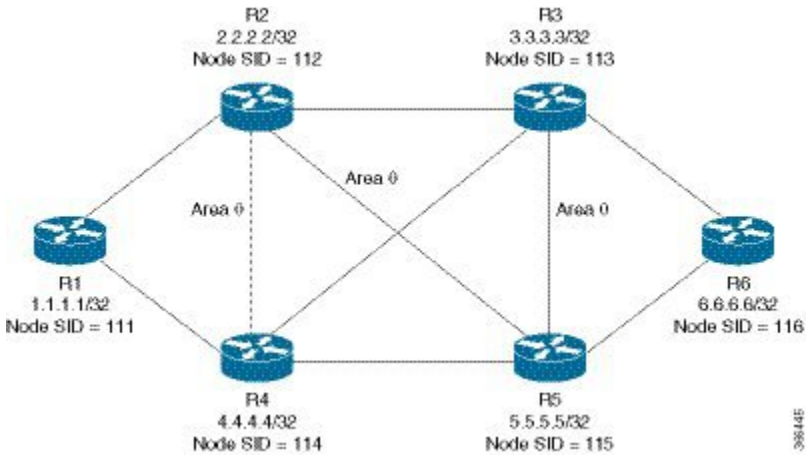
Configuring Segment Routing Traffic Engineering With OSPF

Consider the following inter area and intra area use cases for configuring SR-TE with OSPF:

Configuring Intra Area Tunnel

Consider the following topology to configure intra area tunnel:

Figure 12: Intra Area Tunnel



All the routers are configured in the same area, Area 0.

Configuration at the head end router R1:

```
router ospf 10
fast-reroute per-prefix enable prefix-priority low
fast-reroute per-prefix ti-lfa
segment-routing mpls
mpls traffic-eng router-id Loopback1
mpls traffic-eng area 0

interface GigabitEthernet2   //interface connecting to the router 2
ip address 10.101.1.1 255.255.255.0
ip ospf 10 area 0
ip ospf network point-to-point
negotiation auto
mpls traffic-eng tunnels

interface GigabitEthernet4  //interface connecting to the router 4
ip address 10.101.1.1 255.255.255.0
ip ospf 10 area 0
ip ospf network point-to-point
negotiation auto
mpls traffic-eng tunnels

interface loopback1
ip address 10.1.1.1/32
ip ospf 10 area 0
```

Configuration at the tail-end router R6:

```
router ospf 10
fast-reroute per-prefix enable prefix-priority low
fast-reroute per-prefix ti-lfa
segment-routing mpls
mpls traffic-eng area 0
mpls traffic-eng router-id Loopback1
interface GigabitEthernet2   //interface connecting to the router 3
ip address 10.101.2.1 255.255.255.0
ip ospf 10 area 0
ip ospf network point-to-point
negotiation auto
mpls traffic-eng tunnels
```

```
interface GigabitEthernet4  //interface connecting to the router 5
ip address 10.101.2.1 255.255.255.0
ip ospf 10 area 0
ip ospf network point-to-point
negotiation auto
mpls traffic-eng tunnels

interface loopback1
ip address 10.6.6.6/32
ip ospf 10 area 0
```

Explicit Path SR-TE Tunnel 1

Consider tunnel 1 based only on IP addresses:

```
ip explicit-path name IP_PATH1
 next-address 10.2.2.2
 next-address 10.3.3.3
 next-address 10.6.6.6
!
interface Tunnel1
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng path-option 10 explicit name IP_PATH1 segment-routing
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10
end
```

Explicit Path SR-TE Tunnel 2

Consider tunnel 2 based on node SIDs

```
ip explicit-path name IA_PATH
 next-label 114
 next-label 115
 next-label 116
!
interface Tunnel2
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng bandwidth 10000 class-type 1
 tunnel mpls traffic-eng path-option 10 explicit name IA_PATH segment-routing
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10
end
```

Explicit Path SR-TE Tunnel 3

Consider that tunnel 3 is based on a mix of IP addresses and label

```
ip explicit-path name MIXED_PATH enable
 next-address 10.2.2.2
 next-address 10.3.3.3
 next-label 115
 next-label 116
```

```
!
interface Tunnel3
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng path-option 10 explicit name MIXED_PATH segment-routing
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10
```

Note In the case of mixed path, IP next-hop cannot be used after using Node SIDs in the path. The following path will not be valid:

```
ip explicit-path name MIXED_PATH enable
next-label 115
next-label 116
next-address 10.2.2.2
```

Dynamic Path SR-TE Tunnel 4

Consider that tunnel 4is based on adjacency SIDs

```
interface Tunnel4
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng bandwidth 10000 class-type 1
 tunnel mpls traffic-eng path-option 10 dynamic segment-routing
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10
end
```

Dynamic Path SR-TE Tunnel 5

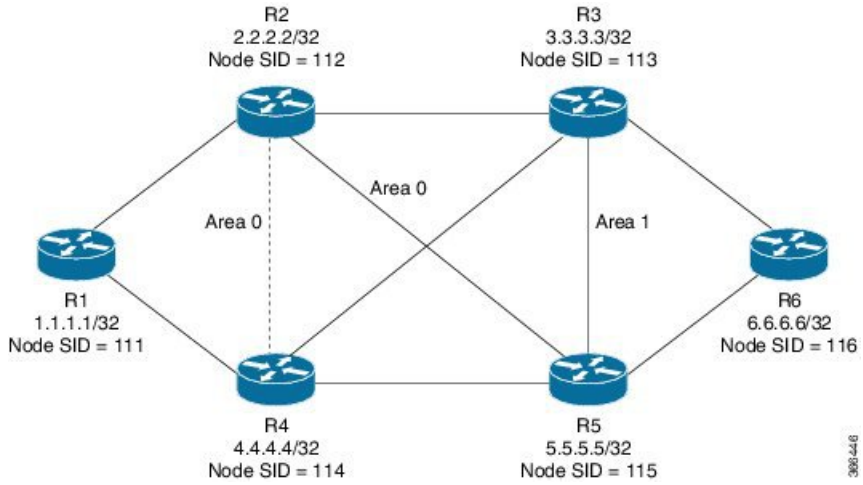
Consider that tunnel 5 is based on Node SIDs

```
 interface Tunnel5
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.6.6.6
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng path-option 10 segment-routing
 tunnel mpls traffic-eng path-selection metric igp
 tunnel mpls traffic-eng load-share 10
```

Configuring Inter Area Tunnel

Consider the following topology to configure inter area tunnel:

Figure 13: Inter Area Tunnel



All the routers are configured in the same area, area 0 except R6 which is configured in area 1.

Configuration at the head end router R1:

```
router ospf 10
fast-reroute per-prefix enable prefix-priority low
fast-reroute per-prefix ti-lfa
segment-routing mpls
mpls traffic-eng router-id Loopback1
mpls traffic-eng area 0

interface GigabitEthernet2   //interface connecting to the router 2
ip address 10.101.1.1 255.255.255.0
ip ospf 10 area 0
ip ospf network point-to-point
negotiation auto
mpls traffic-eng tunnels

interface GigabitEthernet4  //interface connecting to the router 4
ip address 10.101.1.1 255.255.255.0
ip ospf 10 area 0
ip ospf network point-to-point
negotiation auto
mpls traffic-eng tunnels

interface loopback1
ip address 10.1.1.1/32
ip ospf 10 area 0
```

Configuration at the tail-end router R6:

```
router ospf 10
fast-reroute per-prefix enable prefix-priority low
fast-reroute per-prefix ti-lfa
segment-routing mpls
mpls traffic-eng area 1
mpls traffic-eng router-id Loopback1
interface GigabitEthernet2   //interface connecting to the router 3
ip address 10.101.2.1 255.255.255.0
ip ospf 10 area 1
ip ospf network point-to-point
```

```
negotiation auto
mpls traffic-eng tunnels

interface GigabitEthernet4  //interface connecting to the router 5
ip address 10.101.2.1 255.255.255.0
ip ospf 10 area 1
ip ospf network point-to-point
negotiation auto
mpls traffic-eng tunnels

interface loopback1
ip address 10.6.6.6/32
ip ospf 10 area 1
```

Restrictions for Configuring Inter Area Tunnel

The following are the restrictions for configuring inter area tunnel:

- The dynamic option with node and adjacency SID are not supported.
- You can configure inter are tunnel using the explicit path containing only labels and/or IP address and labels.

Note The IP address can be used only be till the Area Border Router (ABR) and after that you need to specify only the labels.

Explicit Path SR-TE Tunnel 1

Consider tunnel 2 is based on node SIDs.

```
ip explicit-path name IA_PATH
next-label 114
next-label 115
next-label 116
!
interface Tunnel2
ip unnumbered Loopback1
tunnel mode mpls traffic-eng
tunnel destination 10.6.6.6
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 6 6
tunnel mpls traffic-eng bandwidth 10000 class-type 1
tunnel mpls traffic-eng path-option 10 explicit name NODE_PATH segment-routing
tunnel mpls traffic-eng path-selection metric igp
tunnel mpls traffic-eng load-share 10
end
```

Explicit Path SR-TE Tunnel 2

Consider that tunnel 3 is based on a mix of IP Addresses and label.

```
ip explicit-path name MIXED_PATH enable
next-address 10.2.2.2
next-address 10.3.3.3
next-label 115
next-label 116
!
```

```
interface Tunnel3
ip unnumbered Loopback1
tunnel mode mpls traffic-eng
tunnel destination 10.6.6.6
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 6 6
tunnel mpls traffic-eng path-option 10 explicit name MIXED_PATH segment-routing
tunnel mpls traffic-eng path-selection metric igp
tunnel mpls traffic-eng load-share 10
```

Verifying Configuration of the SR-TE Tunnels

Use the **show mpls traffic-eng tunnels** *tunnel-number* command to verify the configuration of the SR-TE tunnels.

Verifying Tunnel 1

```
Name: R1_t1                           (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type explicit IP_PATH (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours
      Time since path change: 2 seconds
      Number of LSP IDs (Tun_Instances) used: 1814
    Current LSP: [ID: 1814]
      Uptime: 2 seconds
      Selection: reoptimization
    Prior LSP: [ID: 1813]
      ID: path option unknown
      Removal Trigger: configuration changed
  Tun_Instance: 1814
  Segment-Routing Path Info (ospf 10  area 0)
    Segment0[Node]: 10.4.4.4, Label: 114
    Segment1[Node]: 10.5.5.5, Label: 115
    Segment2[Node]: 10.6.6.6, Label: 116
```

•

Verifying Tunnel 2

```
Name: R1_t2                           (Tunnel1) Destination: 10.6.6.6
```

```
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type explicit IA_PATH (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours, 1 minutes
      Time since path change: 1 seconds
      Number of LSP IDs (Tun_Instances) used: 1815
    Current LSP: [ID: 1815]
      Uptime: 1 seconds
    Prior LSP: [ID: 1814]
      ID: path option unknown
      Removal Trigger: configuration changed
  Tun_Instance: 1815
  Segment-Routing Path Info (ospf 10  area 0)
    Segment0[ - ]: Label: 114
    Segment1[ - ]: Label: 115
    Segment2[ - ]: Label: 116
```

Verifying Tunnel 3

```
Name: R1_t3                           (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type explicit MIXED_PATH (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours, 2 minutes
      Time since path change: 2 seconds
      Number of LSP IDs (Tun_Instances) used: 1816
    Current LSP: [ID: 1816]
      Uptime: 2 seconds
      Selection: reoptimization
    Prior LSP: [ID: 1815]
      ID: path option unknown
      Removal Trigger: configuration changed
  Tun_Instance: 1816
  Segment-Routing Path Info (ospf 10  area 0)
```

```
  Segment0[Node]: 10.2.2.2, Label: 112
  Segment1[Node]: 10.3.3.3, Label: 113
  Segment2[ - ]: Label: 115
  Segment3[ - ]: Label: 116
```

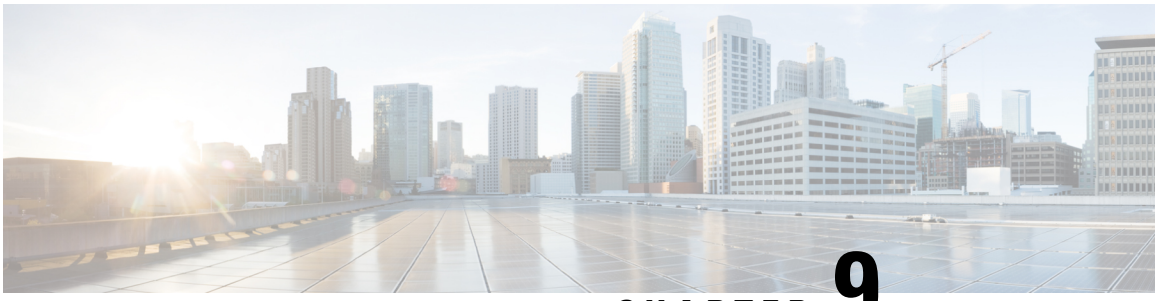
Verifying Tunnel 4

```
Name: R1_t4                             (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) type dynamic (Basis for Setup, path weight 30)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: dynamic path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
      Time since created: 6 days, 19 hours
      Time since path change: 2 seconds
      Number of LSP IDs (Tun_Instances) used: 1813
    Current LSP: [ID: 1813]
      Uptime: 2 seconds
    Prior LSP: [ID: 1806]
      ID: path option unknown
      Removal Trigger: configuration changed
  Tun_Instance: 1813
  Segment-Routing Path Info (ospf 10  area 0)
    Segment0[Link]: 192.168.2.1 - 192.168.2.2, Label: 17
    Segment1[Link]: 192.168.4.2 - 192.168.4.1, Label: 25
    Segment2[Link]: 192.168.8.1 - 192.168.8.2, Label: 300
```

Verifying Tunnel 5

```
Name: R1_t5                             (Tunnel1) Destination: 10.6.6.6
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, type segment-routing (Basis for Setup)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 6  6   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-invalidation timeout: 45000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 10 [200000000]
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: segment-routing path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  History:
    Tunnel:
```

```
    Time since created: 6 days, 19 hours, 4 minutes
    Time since path change: 14 seconds
    Number of LSP IDs (Tun_Instances) used: 1817
  Current LSP: [ID: 1817]
    Uptime: 14 seconds
    Selection: reoptimization
  Prior LSP: [ID: 1816]
    ID: path option unknown
    Removal Trigger: configuration changed
Tun_Instance: 1817
Segment-Routing Path Info (ospf 10  area 0)
  Segment0[Node]: 10.6.6.6, Label: 116
```



CHAPTER 9

BGP Dynamic Segment Routing Traffic Engineering

Border Gateway Protocol (BGP) has become a popular choice as a routing protocol in Data Center (DC) network. The ability to setup Segment Routing-Traffic Engineering (SR-TE) path initiated by BGP simplifies DC network operation.

- Feature Information for BGP Dynamic Segment Routing Traffic Engineering, on page 107
- Restrictions for Segment Routing –Traffic-Engineering Dynamic BGP, on page 107
- Information About Segment Routing –Traffic-Engineering Dynamic BGP, on page 108
- How to Configure TE Label Switched Path Attribute-Set, on page 109

Feature Information for BGP Dynamic Segment Routing Traffic Engineering

Table 7: Feature Information for BGP Dynamic Segment Routing Traffic Engineering

Feature Name	Releases	Feature Information
BGP Dynamic Segment Routing Traffic Engineering	Cisco IOS XE Amsterdam 17.3.2	In BGP dynamic SR-TE, the label Switched Path (LSP) is enabled on demand when defined criteria and policies are met. The following commands were introduced or modified: **mpls traffic-eng lsp attribute** *name*

Restrictions for Segment Routing –Traffic-Engineering Dynamic BGP

- For Anycast SID support to work BGP-TE should be configured with the prepend feature.
- In the case of BGP Dynamic SR-TE if SR-TE fails, forwarding gets broken.

Information About Segment Routing –Traffic-Engineering Dynamic BGP

In BGP dynamic SR-TE, the label Switched Path (LSP) is enabled on demand when defined criteria and policies are met and that is the key difference between manually enabled SR-TE and BGP dynamic SR-TE. Policies, for example, low latency path, minimum cost path, and so on are carried via BGP and matches on a given customer prefix. SR-TE tunnel used for L3VPN or Virtual Private LAN Services (VPLS) using BGP for auto-discovery and signaling is referred to as BGP-TE Dynamic.

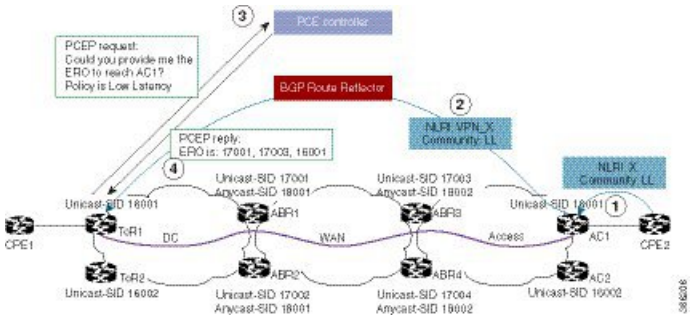
BGP SR-TE dynamic assumes the on-demand auto-tunnel resides in single IGP domain. In this case path computation is done via IGP. SR-TE auto-tunnel created based on the request from BGP is a dynamic SR-TE tunnel. In other words, tunnel path information, or label stack, is computed based on the BGP next-hop and TE attribute configuration. BGP dynamic SR-TE functions to trigger an On-demand LSP (auto-tunnel). The functions include:

- Tag customer prefixes (IPv4 or L3VPN VRF) using communities (community list) via route map configuration.
- Associate each community with a TE attribute-set or profile.

SR-TE profile is locally configured in attribute-set to define certain SR-TE parameters, for example, latency, disjoint path and so on. Once the BGP customer prefixes are mapped to an SR-TE-profile, a tunnel is dynamically created (auto-tunnel or On demand Label Switched Path (LSP)) using the parameters defined in the attribute-set, for each specified BGP next-hop and attribute-set pair associated with the prefixes. A binding SID is associated with each SR-TE auto-tunnel and passed to BGP. The binding SID or binding label is installed into Routing Information Base (RIB) and Forwarding Information Base (FIB). FIB resolves BGP path via the binding SID or binding label, which forwards over the On demand SR-TE auto-tunnel. The binding-SID is also used to steer the customer traffic over the SR-TE LSP.

It must be noted that BGP only carries the SR-TE policy in this case, while path computation is done via IGP in a single IGP domain. In a single IGP domain the headend node has full visibility of the end to end path and the topology engineering database (Traffic Engineering Database or TED). Also it is assumed with BGP Dynamic SR-TE that all the nodes reside within single AS and single IGP domain.

Figure 14: BGP-TE Dynamic Workflow



The above figure depicts the workflow for BGP-TE dynamic using multiple routing domains use case:

1. Customer premise equipment 2 (CPE) sends BGP update for Prefix-X and adds LL community, for example, 100:333.
2. AC1 announces a VPN route for prefix X with LL community.

3. After receiving BGP update of the VPN route matching community LL, ToR1 sends a request to PCE controller for LSP path towards AC1 with low latency TE policy.
4. Path calculation element (PCE) controller replies with a label stack, for example, 17003, 1600.
5. ToR1 creates SR-TE auto-tunnel and installs the route for Prefix-X in VRF of this VPN.

TE Label Switched Path Attribute-Set

TE-LSP attribute-set is used to configure the properties of a LSP. It describes TE profile or policy such as bandwidth, affinities inclusion and exclusion, links/nodes/SRLG inclusion and exclusion, metrics, path disjoint degree and group, and so on that are used to create an auto-tunnel.

How to Configure TE Label Switched Path Attribute-Set

Configuring TE Label Switched Path Attribute-Set

You can use the command **mpls traffic-eng lsp attribute** <*name*> to configure TE-LSP attribute. The following options are available:

```
Mpls traffic-eng lsp attribute name
   affinity      Specify attribute flags for links comprising LSP
   lockdown      Lockdown the LSP--disable reoptimization
   priority      Specify LSP priority
```

TE-LSP attribute command can be extended to support configuration for the two options **pce** and **path-selection**. It can be configured as following:

```
mpls traffic-eng lsp attribute name <test>
    path-selection
       metric <te/igp>
       invalidation <time-out> <drop/tear>
       segment-routing adjacency <protected/unprotected>
```

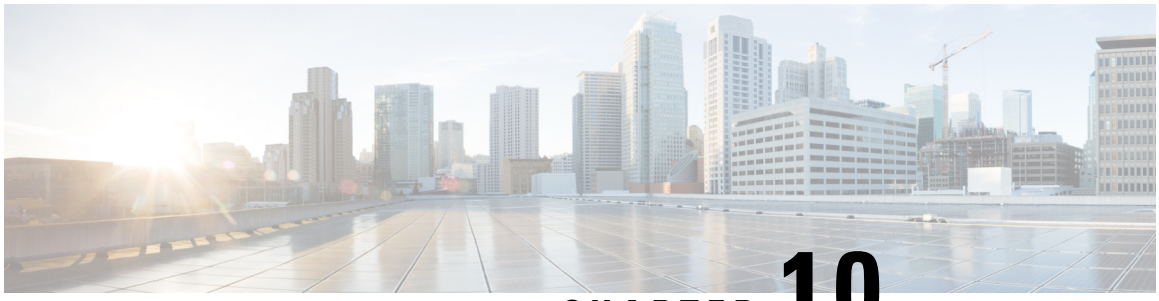
- If pce option is set in the TE attribute the dynamic path is calculated by PCE. Otherwise, the path is calculated locally by TE PCALC (path-calculation) entity. In the later case, IGP has to be configured and the BGP next-hop has to be both advertised by IGP and reachable from the local node over an IGP route.
- The option path-selection metric indicates whether the path calculation is based on TE metrics or IGP metrics. If this option is not configured the global value configured under mpls traffic-eng path-selection metric is used.
- The option **path-selection invalidation** configures the behavior of how an LSP reacts to soft failure from network. When an LSP path has a protected path from IGP against a link or node failure, the failure to the link or node is considered as soft failure.
- The option **path-selection segment-routing adjacency** indicates whether to choose an adjacency-SID with or without IGP protection when calculating LSP label stack.

- The option **pce disjoint-path** indicates the tunnel LSP is a member of disjoint-path group. Any LSPs within the same disjoint-path group do not traverse the same resources, such as links, nodes, or SRLG, in its path. This is used to create two or more tunnel LSPs with disjoint paths.

For BGP-TE Dynamic, a TE attribute name is associated with a BGP route-map set extension as following:

```
route-map <name>
    match community <name>
         set attribute-set  <name>
```

BGP uses the **attribute-set** <*name*> string together with its BGP next-hop to request a SR-TE auto-tunnel.



CHAPTER 10

Segment Routing On Demand Next Hop for L3/L3VPN

When redistributing routing information across domains, provisioning of multi-domain services (L2VPN & L3VPN) has its own complexity and scalability issues. On Demand Next Hop (ODN) triggers delegation of computation of an end-to-end LSP to a PCE controller including constraints and policies without doing any redistribution. It then installs the replied multi-domain LSP for the duration of the service into the local forwarding information base (FIB).

- Feature Information for Segment Routing On Demand Next Hop for L3/L3VPN, on page 111
- Restrictions for Segment Routing On Demand SR PFP ODN AUTO STEERING (PCE DELEGATED) for L3/L3VPN, on page 112
- Information About Segment Routing On Demand SR PFP ODN AUTO STEERING (PCE DELEGATED) for L3/L3VPN, on page 112
- How to Configure Segment Routing On Demand Next Hop for L3/L3VPN, on page 113
- Verifying Segment Routing On Demand Next Hop for L3/L3VPN, on page 117

Feature Information for Segment Routing On Demand Next Hop for L3/L3VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 8: Feature Information for Segment Routing On Demand Next Hop for L3/L3VPN

Feature Name	Releases	Feature Information
Segment Routing On Demand Next Hop for L3/L3VPN	Cisco IOS XE Amsterdam 17.3.2	On-Demand Next Hop (ODN) triggers delegation of computation of an end-to-end LSP to a PCE controller including constraints and policies without doing any redistribution. The following commands were introduced or modified: **route-map BGP_TE_MAP permit**, **mpls traffic-eng tunnels**, **sh bgp li li summary**, **sh pce client peer**, **sh pce ipv4 peer**, **sh ip route vrf sr**, **sh ip bgp vpnv4 vrf sr**, **sh ip cef label-table**, **sh mpls traffic-eng tunnels**, **sh pce client lsp brief**, **sh pce lsp summ**, **sh pce lsp det**, **routing-default-optimize**

Restrictions for Segment Routing On Demand SR PFP ODN AUTO STEERING (PCE DELEGATED) for L3/L3VPN

- On Demand Next Hop (ODN) anycast SID is not supported.
- ODN for IPv6 is not supported.
- SR ODN tunnel is not supported with BGP Nonstop Routing (NSR). It is only supported with BGP Nonstop Forwarding (NSF).

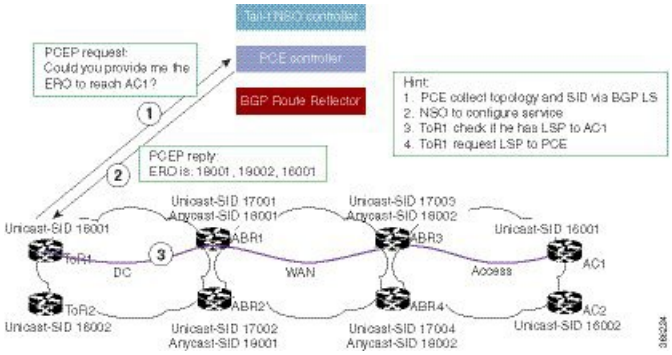
  To enable BGP NSF, use the following command:

```
bgp grace-full restart
neighbor 10.0.0.2 ha-mode graceful-restart
```

Information About Segment Routing On Demand SR PFP ODN AUTO STEERING (PCE DELEGATED) for L3/L3VPN

On Demand SR PFP ODN AUTO STEERING (PCE DELEGATED) leverages upon BGP Dynamic SR-TE capabilities and adds the path computation (PCE) ability to find and download the end to end path based on the requirements. ODN triggers an SR-TE auto-tunnel based on the defined BGP policy. As shown in the below figure, an end to end path between ToR1 and AC1 can be established from both ends based on low latency or other criteria for VRF (L3VPN) or IPv4 services. The work-flow for ODN is summarized as follows:

Figure 15: ODN Operation



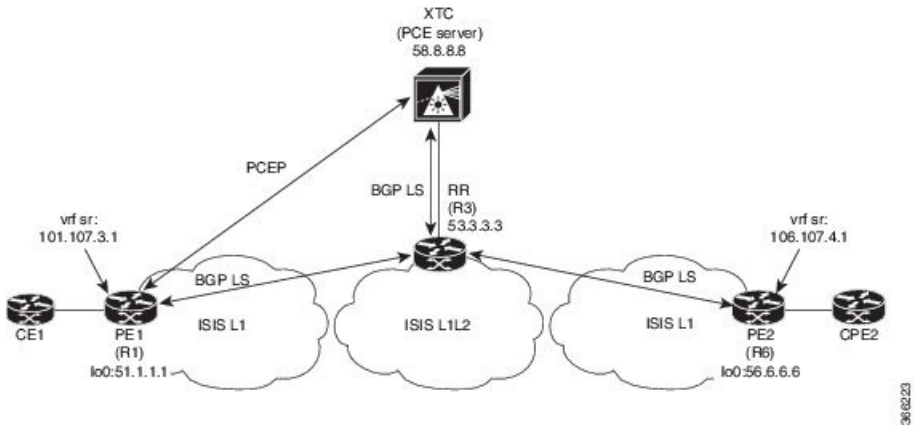
1. PCE controller collects topology and SIDs information via BGP Link State (BGP-LS). For more information on BGP-LS, refer BGP Link-State.
2. If NSO controller is enable, it configures L3VPN VRF or IPv4 prefixes and requests are sent to ToR1 and AC1.
3. ToR1 and AC1 checks if a LSP towards each other exists. If not, a request is sent to the PCE controller to compute that SR-TE path that matches SR-TE policy that is carried via BGP.
4. PCE controller computes the path and replies with a label stack (18001, 18002, 16001, example in ToR1).
5. ToR1 and AC1 create a SR-TE auto-tunnel and reply back to the NSO controller indicating that the LSP for VRF or IPv4 is up and operational.

How to Configure Segment Routing On Demand Next Hop for L3/L3VPN

Configuring Segment Routing On Demand Next Hop for L3/L3VPN

Perform the following steps to configure on-demand next hop for SR-TE. The below figure is used as a reference to explain the configuration steps.

Figure 16: ODN Auto-Tunnel Setup



1. Configure the router (R6 tail end) with VRF interface.

```
interface GigabitEthernet0/2/2
 vrf forwarding sr
 ip address 10.0.0.1 255.0.0.0
 negotiation auto

interface Loopback0
 ip address 192.168.0.1 255.255.0.0
 ip router isis 1
```

2. Tags VRF prefix with BGP community on R6 (tail end).

```
route-map BGP_TE_MAP permit 9
 match ip address traffic
 set community 3276850

ip access-list extended traffic
 permit ip 10.0.0.1 255.255.0.0 any
```

3. Enable BGP on R6 (tail end) and R1 (head end) to advertise and receive VRF SR prefix and match on community set on R6 (tail end).

```
 router bgp 100
 bgp router-id 172.16.0.1
 bgp log-neighbor-changes
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 10.0.0.2 remote-as 100
 neighbor 10.0.0.2 update-source Loopback0

 address-family ipv4
  neighbor 10.0.0.2 activate
  neighbor 10.0.0.2 send-community both
  neighbor 10.0.0.2 next-hop-self
 exit-address-family

 address-family vpnv4
  neighbor 10.0.0.2 activate
  neighbor 10.0.0.2 send-community both
  neighbor 10.0.0.2 route-map BGP_TE_MAP out
 exit-address-family
```

```
 address-family link-state link-state
  neighbor 10.0.0.2 activate
 exit-address-family

 address-family ipv4 vrf sr
  redistribute connected
 exit-address-family

route-map BGP_TE_MAP permit 9
 match ip address traffic
 set community 3276850

ip access-list extended traffic
 permit ip 10.0.0.1 255.255.0.0 any


router bgp 100
 bgp router-id 192.168.0.2
 bgp log-neighbor-changes
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 10.0.0.2 remote-as 100
 neighbor 10.0.0.2 update-source Loopback0


address-family ipv4
  neighbor 10.0.0.2 activate
  neighbor 10.0.0.2 send-community both
  neighbor 10.0.0.2 next-hop-self
 exit-address-family

 address-family vpnv4
  neighbor 10.0.0.2 activate
  neighbor 10.0.0.2 send-community both
  neighbor 10.0.0.2 route-map BGP_TE_MAP in
 exit-address-family

 address-family link-state link-state
  neighbor 10.0.0.2 activate
 exit-address-family

 address-family ipv4 vrf sr
  redistribute connected
 exit-address-family

route-map BGP_TE_MAP permit 9
 match community 1
 set attribute-set BGP_TE5555

ip community-list 1 permit 3276850

mpls traffic-eng lsp attributes BGP_TE5555
 path-selection metric igp
 pce
```

4. Enable PCE and auto-tunnel configurations on R1.

```
mpls traffic-eng tunnels
mpls traffic-eng pcc peer 10.0.0.3 source 10.0.0.4 precedence 255
mpls traffic-eng auto-tunnel p2p tunnel-num min 2000 max 5000
```

5. Enable all core links with SR-TE configurations and ensure that they are enabled as point to point interfaces.

```
mpls traffic-eng tunnels

interface GigabitEthernet0/2/0
 ip address 10.102.6.1 255.255.255.0
 ip router isis 1
 mpls traffic-eng tunnels
 isis network point-to-point

interface GigabitEthernet0/3/1
 vrf forwarding sr
 ip address 10.107.3.1 255.255.255.0
 negotiation auto

end
```

6. Enable R3 (RR) to advertise TED to the PCE server via BGP-LS.

```
router isis 1
 net 49.0002.0000.0000.0003.00
 ispf level-1-2
 metric-style wide
 nsf cisco
 nsf interval 0
 distribute link-state
 segment-routing mpls
 segment-routing prefix-sid-map advertise-local
 redistribute static ip level-1-2
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
 mpls traffic-eng level-2

router bgp 100
 bgp router-id 10.0.0.2
 bgp log-neighbor-changes
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 10.0.0.3 remote-as 100
 neighbor 10.0.0.3 update-source Loopback0

 address-family ipv4
 neighbor 10.0.0.3 activate
 exit-address-family
```

7. Enable PCE server configuration and verify BGP-LS session is properly established with RR.

```
Device# sh bgp li li summary
BGP router identifier 10.0.0.3, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 1436
BGP main routing table version 1436
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs
BGP is operating in STANDALONE mode.
Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker            1436       1436          1436       1436           1436
       0

Neighbor        Spk    AS MsgRcvd   MsgSent   TblVer  InQ  OutQ  Up/Down  St/PfxRcd
10.0.0.2          0      100   19923       17437        1436       0     0
1w2d        103
```

```
Device# sh pce ipv4 topo | b Node 3
Node 3
  TE router ID: 10.0.0.2
  Host name: R3
  ISIS system ID: 0000.0000.0003 level-1

  ISIS system ID: 0000.0000.0003 level-2
  Prefix SID:
    Prefix 10.0.0.2, label 20011 (regular)
```

Verifying Segment Routing On Demand Next Hop for L3/L3VPN

The ODN verifications are based on L3VPN VRF prefixes.

1. Verify that PCEP session between R1 (headend and PCE server) is established.

```
Device# sh pce client peer
PCC's peer database:
--------------------
Peer address: 10.0.0.3 (best PCE)
  State up
  Capabilities: Stateful, Update, Segment-Routing
```

2. Verify that PCEP session is established between all the peers (PCCs).

```
Device# sh pce ipv4 peer
PCE's peer database:
--------------------
Peer address: 10.0.0.4
  State: Up
  Capabilities: Stateful, Segment-Routing, Update
Peer address: 172.16.0.5
  State: Up
  Capabilities: Stateful, Segment-Routing, Update
```

3. Verify that R1 (headend) has no visibility to R6 loopback address.

```
Device# sh ip route 192.168.0.1
% Network not in table
```

4. Verify that VRF prefix is injected via MP-BGP in R1 VRF SR routing table.

```
Device# sh ip route vrf sr
Routing Table: sr
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
Gateway of last resort is not set
     10.0.0.6/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.7/24 is directly connected, GigabitEthernet0/3/1
L       10.0.0.7/32 is directly connected, GigabitEthernet0/3/1
      10.0.0.8/24 is subnetted, 1 subnets
B       10.0.0.9 [200/0] via binding label: 865, 4d21h
```

5. Verify that BGP is associating properly the policy and binding SID with the VRF prefix.

```
Device# sh ip bgp vpnv4 vrf sr 10.107.4.0
BGP routing table entry for 100:100:10.107.4.0/24, version 3011
Paths: (1 available, best #1, table sr)
  Not advertised to any peer
  Refresh Epoch 4
  Local
    192.168.0.1 (metric 10) (via default) from 10.0.0.2 (10.0.0.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Community: 3276850
      Extended Community: RT:100:100
      Originator: 192.168.0.1, Cluster list: 10.0.0.2
      mpls labels in/out nolabel/1085
      binding SID: 865 (BGP_TE5555)
      rx pathid: 0, tx pathid: 0x0
```

6. Verify binding label association with VRF prefix.

```
Device# sh ip route vrf sr 10.107.4.0
Routing Table: sr
Routing entry for 10.107.4.0/24
  Known via "bgp 100", distance 200, metric 0, type internal
  Routing Descriptor Blocks:
  *  Binding Label: 865, from 10.0.0.2, 4d22h ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: 1085
      MPLS Flags: NSF
```

7. Verify that VRF prefix is forwarded via ODN auto-tunnel.

```
Device# sh ip cef label-table
Label               Next Hop             Interface
0                    no route
865                  attached             Tunnel2000

     Device# sh ip cef vrf sr 10.107.4.0 detail
10.0.0.8/24, epoch 15, flags [rib defined all labels]
  recursive via 865 label 1085
    attached to Tunnel2000
```

8. Verify ODN auto-tunnel status.

```
Device# sh mpls traffic-eng tunnels
P2P TUNNELS/LSPs:
Name: R1_t2000                 (Tunnel2000) Destination: 192.168.0.1 Ifhandle: 0x6F5
(auto-tunnel for BGP TE)
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected---
auto-tunnel 2000
    path option 1, (SEGMENT-ROUTING) (PCE) type dynamic (Basis for Setup, path weight
 10)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: IGP (interface)
    Path Selection:
     Protection: any (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
    Hop Limit: disabled
    Cost Limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear
    AutoRoute: disabled LockDown: disabled Loadshare: 0 [0] bw-based
    auto-bw: disabled
    Attribute-set: BGP_TE5555--- attribute-set
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
```

```
  Active Path Option Parameters:
    State: dynamic path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  PCEP Info:
    Delegation state: Working: yes   Protect: no
    Working Path Info:
      Request status: processed
      Created via PCRep message from PCE server: 10.0.0.3--□ via PCE server
      PCE metric: 30, type: IGP
    Reported paths:
      Tunnel Name: Tunnel2000_w
       LSPs:
        LSP[0]:
         source 10.0.0.4, destination 192.168.0.1, tunnel ID 2000, LSP ID 1
         State: Admin up, Operation active
         Binding SID: 865
         Setup type: SR
         Bandwidth: requested 0, used 0
         LSP object:
           PLSP-ID 0x807D0, flags: D:0 S:0 R:0 A:1 O:2
         Metric type: IGP, Accumulated Metric 0
         ERO:
           SID[0]: Adj, Label 2377, NAI: local 10.102.6.1 remote 10.0.0.10
           SID[1]: Unspecified, Label 17, NAI: n/a
           SID[2]: Unspecified, Label 20, NAI: n/a
  History:
    Tunnel:
      Time since created: 4 days, 22 hours, 21 minutes
      Time since path change: 4 days, 22 hours, 21 minutes
      Number of LSP IDs (Tun_Instances) used: 1
    Current LSP: [ID: 1]
      Uptime: 4 days, 22 hours, 21 minutes
  Tun_Instance: 1
  Segment-Routing Path Info (isis  level-1)
    Segment0[Link]: 10.102.6.1 - 10.0.0.10, Label: 2377
    Segment1[ - ]: Label: 17
    Segment2[ - ]: Label: 20
```

9. Verify ODN auto-tunnel LSP status on R1 (headend).

```
Device# sh pce client lsp brief
PCC's tunnel database:
----------------------
 Tunnel Name: Tunnel2000_w
   LSP ID 1
 Tunnel Name: Tunnel2000_p

R1# sh pce client lsp detail
PCC's tunnel database:
----------------------
Tunnel Name: Tunnel2000_w
 LSPs:
  LSP[0]:
   source 10.0.0.4, destination 192.168.0.1, tunnel ID 2000, LSP ID 1
   State: Admin up, Operation active
   Binding SID: 865
   Setup type: SR
   Bandwidth: requested 0, used 0
   LSP object:
     PLSP-ID 0x807D0, flags: D:0 S:0 R:0 A:1 O:2
   Metric type: IGP, Accumulated Metric 0
   ERO:
     SID[0]: Adj, Label 2377, NAI: local 10.102.6.1 remote 10.0.0.10
     SID[1]: Unspecified, Label 17, NAI: n/a
     SID[2]: Unspecified, Label 20, NAI: n/a
```

10. Verify ODN LSP status on the PCE server.

```
Device# sh pce lsp summ

PCE's LSP database summary:
--------------------------------
All peers:
 Number of LSPs:         1
  Operational: Up:       1 Down:                 0
  Admin state: Up:       1 Down:                 0
  Setup type: RSVP:      0 Segment routing:      1


Peer 10.0.0.4:
 Number of LSPs:         1
  Operational: Up:       1 Down:                 0
  Admin state: Up:       1  Down:                 0
  Setup type: RSVP:       0 Segment routing:      1
```

11. Verify detailed LSP information on the PCE server.

```
Device# sh pce lsp det
PCE's tunnel database:
----------------------
PCC 10.0.0.4:
Tunnel Name: Tunnel2000_w
 LSPs:
  LSP[0]:
   source 10.0.0.4, destination 192.168.0.1, tunnel ID 2000, LSP ID 48
   State: Admin up, Operation active
   Binding SID: 872
   PCEP information:
     plsp-id 526288, flags: D:1 S:0 R:0 A:1 O:2
   Reported path:
     Metric type: IGP, Accumulated Metric 0
      SID[0]: Adj, Label 885, Address: local 10.0.0.9 remote 10.0.0.10
      SID[1]: Unknown, Label 17,
      SID[2]: Unknown, Label 20,
   Computed path:
     Computed Time: Tue Dec 20 13:12:57 2016 (00:11:53 ago)
     Metric type: IGP, Accumulated Metric 30
      SID[0]: Adj, Label 885, Address: local 10.0.0.9 remote 10.0.0.10
      SID[1]: Adj, Label 17, Address: local 10.0.0.12 remote 10.0.0.13
      SID[2]: Adj, Label 20, Address: local 10.0.0.14 remote 10.0.0.14
   Recorded path:
     None
```

12. Shutdown the interface that is connected to VRF SR so that the prefix is no longer advertised by MP-BGP.

```
Device# int gig0/2/2
Device(config-if)#shut
```

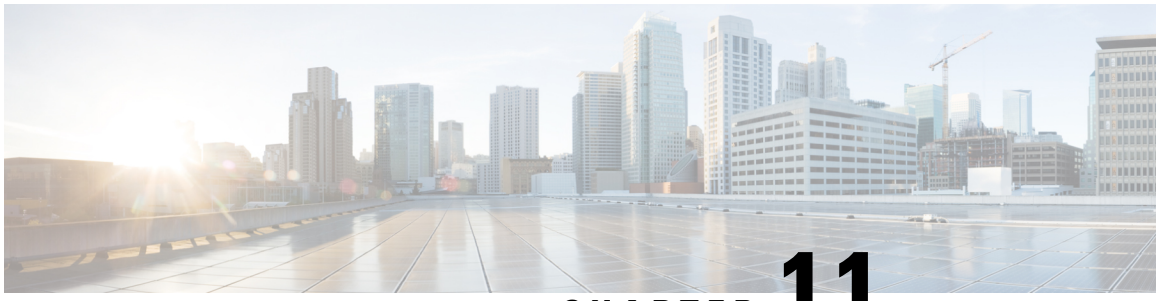
13. Verify that VRF prefix is no longer advertised to R1 (headend) via R6 (tailend).

```
Device# sh ip route vrf sr
Routing Table: sr
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
```

```
Gateway of last resort is not set
      10.0.0.6/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.0.7/24 is directly connected, GigabitEthernet0/3/1
L        10.0.0.8/32 is directly connected, GigabitEthernet0/3/1
```

14. Verify that no ODN auto-tunnel exists.

```
Device# sh mpls traffic-eng tunnels
P2P TUNNELS/LSPs:
P2MP TUNNELS:
P2MP SUB-LSPS:
```



CHAPTER 11

Segment Routing On Demand for L2VPN/VPWS

On-Demand Next Hop (ODN) for Layer 2 Virtual Private Network (L2VPN) creates a segment routing (SR) traffic-engineering (TE) auto-tunnel and uses the auto-tunnel for pseudowire dataplane.

- Feature Information for Segment Routing On Demand Next Hop for L2VPN/VPWS, on page 123
- Restrictions for Segment Routing On Demand Next Hop for L2VPN/VPWS, on page 124
- Information About Segment Routing On Demand Next Hop for L2VPN/VPWS, on page 124
- How to Configure Segment Routing On Demand Next Hop for L2VPN/VPWS, on page 125
- Configuring Segment Routing On Demand Next Hop for L2VPN/VPWS With Prepend Option, on page 127
- Configuring Preferred Path for Segment Routing On Demand Next Hop for L2VPN/VPWS, on page 127
- Configuring Autoroute Destination for Segment Routing On Demand Next Hop for L2VPN/VPWS, on page 128
- Verifying Segment Routing On Demand Next Hop for L2VPN/VPWS , on page 128

Feature Information for Segment Routing On Demand Next Hop for L2VPN/VPWS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access the Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 9: Feature Information for Segment Routing On Demand Next Hop for L2VPN/VPWS

Feature Name	Releases	Feature Information
Segment Routing On Demand Next Hop for L2VPN/VPWS	Cisco IOS XEAmsterdam 17.3.2	ODN for L2VPN is to create a SR TE auto-tunnel and use the auto-tunnel for pseudo-wire data-plane. The peer IP address is the destination of tunnel and TE LSP attribute determines the path of the tunnel. The following commands were added or modified: **sh mpls l2 vc**, **sh mpls l2 vc detail**, **sh l2vpn atom preferred-path**, **sh l2vpn atom vc**, **sh mpl traffic-eng tun tun 2000**, **sh mpls ldp discovery**, **sh mpl ldp nei**, **sh int pseudowire 4243**, **sh xconnect all**.

Restrictions for Segment Routing On Demand Next Hop for L2VPN/VPWS

- Layer-2 VPN/VPWS (Virtual Private Wire Service) On Demand Next Hop (ODN) is not supported with peudowire (PW) class.
- The segment routing on demand for L2VPN or VPWS is not supported for BGP signaled/ADVPWS or Virtual Private LAN Service (VPLS).
- Only Segment-Routing TE tunnels are supported and created for L2VPN using attribute-set.
- L2VPN preferred path bandwidth related configuration does not take effect when TE attribute-set is configured.
- Only L2-VPN ODN VPWS with LDP signaling is supported.

Information About Segment Routing On Demand Next Hop for L2VPN/VPWS

On Demand Next Hop (ODN) for L2VPN creates an SR TE auto-tunnel and uses the auto-tunnel for pseudowire dataplane. The peer IP address is the destination of tunnel and TE LSP attribute determines path of the tunnel. Sometimes a pseudowire connection may need to span multiple interior gateway protocol (IGP) areas while LDP is used as signaling protocol. The pseudowire endpoint provider edge's (PE) loopback addresses are not distributed across IGP area boundaries. In this case, one PE may not have a default route (or an exact match route) in its RIB to reach the peer PE of the pseudowire connection. Thus the pseudowire connection can not be signaled by LDP. A new option **autoroute destination** is introduced under LSP attribute to address this problem. When a LSP attribute is configured using the **autoroute destination** command, auto-tunnel uses the LSP attribute to automatically create a static route for the tunnel destination with the auto-tunnel interface as the next hop. This static route enables LDP to establish a LDP a session and exchange label mapping messages between two pseudowire endpoints.

Note Use the autoroute destination command only to configure LSP attribute used by LDP signaled L2VPN. It is not needed for BGP signaled Layer-3 VPN ODN.

AToM Manager

Any Transport over MPLS (AToM) manager maintains a database of auto-tunnels on a pair of attribute set and peer ip addresses, the AToM manager can add or delete an SR TE auto-tunnel for a pseudowire interface (VC).

Any VC that is configured with the same attribute-set or peer uses the same auto-tunnel. An auto-tunnel can be removed from the database using TE service if an attribute set or peer pair is no longer used by any pseudowire interfaces.

Inter-Area L2VPN ODN

When LDP is used as a signaling protocol and pesudowire connection is spanned across multiple Interior Gateway Protocols (IGPs), the pseudowire endpoint PE's loopback addresses are not distributed across IGP area boundaries. In this case, one PE may not have a default route (or an exact match route) in its RIB to reach the peer PE of the pseudowire connection. Thus the pseudowire connection can not be signaled by LDP.

How to Configure Segment Routing On Demand Next Hop for L2VPN/VPWS

You can use either pesudowire interface command or template method to configure L2VPN/VPWS.

Configuring Segment Routing On Demand Next Hop for L2VPN/VPWS Using Pesudowire Interface Commands

1. Run the following command on headend node (R1):

```
R1#
!
mpls traffic-eng auto-tunnel p2p tunnel-num min 2000 max 2002
!
interface GigabitEthernet0/3/1
 no ip address
 negotiation auto
 service instance 300 ethernet
  encapsulation dot1q 300
!
interface pseudowire4243
 encapsulation mpls
 neighbor 10.6.6.6 300
 preferred-path segment-routing traffic-eng attribute-set L2VPNODN
!
l2vpn xconnect context foobar
 member GigabitEthernet0/3/1 service-instance 300
```

```
 member pseudowire4243
!
mpls traffic-eng lsp attributes L2VPNODN
 priority 7 7
 path-selection metric te
!
end
```

2. Run the following command at tail end (R2):

```
R2#
!
mpls traffic-eng auto-tunnel p2p tunnel-num min 2000 max 2002

interface pseudowire4243
 encapsulation mpls
 neighbor 10.1.1.1 300
 preferred-path segment-routing traffic-eng attribute-set L2VPNODN
!
interface GigabitEthernet0/2/2
 no ip address
 negotiation auto
 service instance 300 ethernet
  encapsulation dot1q 300
!
l2vpn xconnect context foobar
 member GigabitEthernet0/3/1 service-instance 300
 member pseudowire4243
!
mpls traffic-eng lsp attributes L2VPNODN
 priority 7 7
 path-selection metric te
!
end
```

Configuring Segment Routing On Demand Next Hop for L2VPN/VPWS Using Template Commands

1. Run the following command at headend node (R1):

```
R1#
template type pseudowire test
 encapsulation mpls
 preferred-path segment-routing traffic-eng attribute-set L2VPNODN
!
interface GigabitEthernet0/3/1
 no ip address
 negotiation auto
 service instance 400 ethernet
  encapsulation dot1q 400
!
l2vpn xconnect context foobar2
 member 10.6.6.6 400 template test
 member GigabitEthernet0/3/1 service-instance 400
```

2. Run the following command at tail end (R2):

```
R2#
!
```

```
template type pseudowire test
 encapsulation mpls
 preferred-path segment-routing traffic-eng attribute-set L2VPNODN
!
interface GigabitEthernet0/2/2
 no ip address
 negotiation auto
 service instance 400 ethernet
  encapsulation dot1q 400
!
l2vpn xconnect context foobar2
 member 10.1.1.1 400 template test
 member GigabitEthernet0/2/2 service-instance 400
!
end
```

Configuring Segment Routing On Demand Next Hop for L2VPN/VPWS With Prepend Option

To control the path of LSP it is possible to enable prepend option. The prepend option is only supported with intra-area and supports labeled paths only. To enable prepend option use the following CLI:

```
R1(config-lsp-attr)#path-selection segment-routing prepend
R1(config-lsp-attr-sr-prepend)#?
Segment-routing label prepend commands:
  exit   Exist from segment-routing prepend config mode
  index  Specify the next entry index to add, edit or delete
  list   List all prepend entries
  no     Delete a specific entry index
R1(config-lsp-attr-sr-prepend)#index ?
  <1-10>  Entry index number
  last-hop    Indicates the end of label list
  next-label  Specify the next MPLS label in the path
```

Note If last-hop option indicates tail end node. If this option is only used no control on LSP path can be done.

Configuring Preferred Path for Segment Routing On Demand Next Hop for L2VPN/VPWS

To bring down virtual circuit (VC) in case of LSP failure, which could be either because of path fail or removing a command, disable the fallback mode.

```
preferred-path segment-routing traffic-eng attribute-set L2VPNODN
disable-fallback  disable fall back to alternative route
```

Configuring Autoroute Destination for Segment Routing On Demand Next Hop for L2VPN/VPWS

For inter-area destination, IP address may not be installed at headend. You need to have destination IP address installed to enable a targeted LDP session for L2-VPN VPWS. To enable a targeted LDP session for L2VPN VPWS, configure the auto-route destination under the attribute set:

```
Device#
mpls traffic-eng lsp attributes L2VPNODN
 priority 7 7
 path-selection metric te
 pce
 autoroute destination
!
end
```

The destination address gets installed via L2-VPN ODN LSP as a static route.

Run the following commands to verify autoroute destination configuration:

```
Device#sh ip route 10.6.6.6
Routing entry for 10.6.6.6/32
  Known via "static", distance 1, metric 0 (connected)
  Routing Descriptor Blocks:
  * directly connected, via Tunnel2000------------------------ L2-VPN ODN LSP
      Route metric is 0, traffic share count is 1


Device#sh mpls for 10.6.6.6
Local      Outgoing             Prefix          Bytes Label      Outgoing   Next Hop
Label      Label               or Tunnel Id     Switched          interface
25           [T]  Pop Label    10.6.6.6/32      0                   Tu2000     point2point
```

Verifying Segment Routing On Demand Next Hop for L2VPN/VPWS

1. **sh mpls l2 vc**

```
Device#sh mpls l2 vc
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Gi0/3/1        Eth VLAN 300               10.6.6.6        300        UP
```

2. **sh mpls l2 vc detail**

```
Device# sh mpls l2 vc detail
Local interface: Gi0/3/1 up, line protocol up, Eth VLAN 300 up
  Interworking type is Ethernet
  Destination address: 10.6.6.6, VC ID: 300, VC status: up
    Output interface: Tu2000, imposed label stack {23 17 20}----- 20 is the VC label
assigned by R6
    Preferred path: Tunnel2000,  active
    Default path: ready
```

```
    Next hop: point2point
  Create time: 00:15:48, last status change time: 00:15:38
    Last label FSM state change time: 00:15:38
  Signaling protocol: LDP, peer 10.6.6.6:0 up
    Targeted Hello: 10.1.1.1(LDP Id) -> 10.6.6.6, LDP is UP
    Graceful restart: not configured and not enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 2032, remote 20
    Group ID: local 20, remote 25
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 10.6.6.6/300, local label: 2032
  Dataplane:
    SSM segment/switch IDs: 10198/6097 (used), PWID: 1001
  VC statistics:
    transit packet totals: receive 0, send 0
    transit byte totals:   receive 0, send 0
    transit packet drops:  receive 0, seq error 0, send 0
```

3. sh l2vpn atom preferred-path

```
Device#  sh l2vpn atom preferred-path
Tunnel interface    Bandwidth Tot/Avail/Resv         Peer ID         VC ID
------------------       ---------------------------------          ---------------
 ----------
Tunnel2000                                                            10.6.6.6
300
!
end
```

4. sh l2vpn atom vc

```
Device# sh l2vpn atom vc
Interface Peer ID         VC ID      Type   Name                       Status
--------- --------------      ----------    ------ ------------------------ ----------
pw4243    10.6.6.6        300        p2p    foobar                          UP
!
end
```

5. sh mpl traffic-eng tun tun 2000

```
Device# sh mpl traffic-eng tun tun 2000
Name: R1_t2000                 (Tunnel2000) Destination: 10.6.6.6 Ifhandle: 0x7EE
(auto-tunnel for atom)
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, (SEGMENT-ROUTING) (PCE) type dynamic (Basis for Setup, path weight
```

```
30)
  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (interface)
    Path Selection:
     Protection: any (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
    Hop Limit: disabled
    Cost Limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear
    AutoRoute: disabled LockDown: disabled Loadshare: 0 [0] bw-based
    auto-bw: disabled
    Attribute-set: L2VPNODN
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: dynamic path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
PCEP Info:
    Delegation state: Working: yes   Protect: no
    Delegation peer: 10.8.8.8
    Working Path Info:
      Request status: processed
      Created via PCRep message from PCE server: 10.8.8.8
      PCE metric: 30, type: TE
    Reported paths:
      Tunnel Name: Tunnel2000_w
       LSPs:
        LSP[0]:
         source 10.1.1.1, destination 10.6.6.6, tunnel ID 2000, LSP ID 4
         State: Admin up, Operation active
         Binding SID: 20
         Setup type: SR
         Bandwidth: requested 0, used 0
         LSP object:
           PLSP-ID 0x807D0, flags: D:0 S:0 R:0 A:1 O:2
         Metric type: TE, Accumulated Metric 30
         ERO:
           SID[0]: Adj, Label 19, NAI: local 10.104.1.1 remote 10.104.1.2
           SID[1]: Adj, Label 23, NAI: local 10.104.12.2 remote 10.104.12.1
           SID[2]: Adj, Label 17, NAI: local 10.106.13.1 remote 10.106.13.2
       PLSP Event History (most recent first):
        Tue Jun 20 10:04:48.514: PCRpt create LSP-ID:4, SRP-ID:0, PST:1, METRIC_TYPE:2,
 REQ_BW:0, USED_BW:0
        Tue Jun 20 10:04:48.511: PCRep RP-ID:9
        Tue Jun 20 10:04:48.505: PCReq RP-ID:9, LSP-ID:4, REQ_BW:0
  History:
    Tunnel:
      Time since created: 18 minutes, 26 seconds
      Time since path change: 17 minutes, 9 seconds
      Number of LSP IDs (Tun_Instances) used: 4
    Current LSP: [ID: 4]
      Uptime: 17 minutes, 9 seconds
  Tun_Instance: 4
  Segment-Routing Path Info (isis  level-2)
    Segment0[Link]: 10.104.1.1 - 10.104.1.2, Label:  19--------□ will not be shown in
sh mpls l2 vc output
    Segment1[Link]: 10.104.12.2 - 10.104.12.1, Label: 23
    Segment2[Link]: 10.106.13.1 - 10.106.13.2, Label: 17
!
end
```

6. **sh mpls ldp discovery**

```
Device# sh mpls ldp discovery
Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Targeted Hellos:
        10.1.1.1 -> 10.6.6.6 (ldp): active/passive, xmit/recv
            LDP Id: 10.6.6.6:0
```

7. **sh mpl ldp nei**

```
Device# sh mpl ldp nei
Peer LDP Ident: 10.6.6.6:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.6.6.6.38574 - 10.1.1.1.646
        State: Oper; Msgs sent/rcvd: 43/42; Downstream
        Up time: 00:19:33
        LDP discovery sources:
          Targeted Hello 10.1.1.1 -> 10.6.6.6, active, passive
        Addresses bound to peer LDP Ident:
          10.106.2.2      10.106.13.2     10.6.6.6
!
```

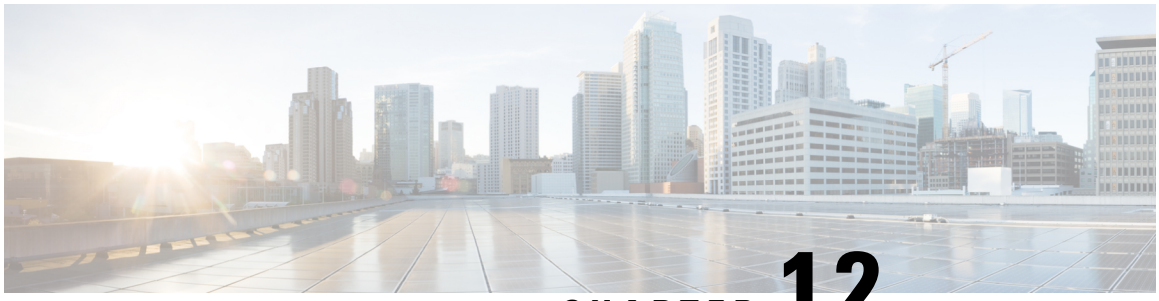
8. **sh int pseudowire 4243**

```
Device# sh int pseudowire 4243
pseudowire4243 is up
    MTU 1500 bytes, BW not configured
    Encapsulation mpls
    Peer IP 10.6.6.6, VC ID 300
    RX    0 packets 0 bytes 0 drops
    TX    0 packets 0 bytes 0 drops
!
```

9. **sh xconnect all**

```
Device# sh xconnect all
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                                    S1 Segment 2
  S2
------+---------------------------------+--+---------------------------------+--
UP pri   ac Gi0/3/1:300(Eth VLAN)        UP mpls 10.6.6.6:300                 UP
```



CHAPTER 12

Fast Convergence Default Optimize

The fast convergence default optimize feature modifies the default settings of all the protocols to recommended defaults for fast convergence.

- Feature Information for Fast Convergence Default Optimize, on page 133
- Information About Fast Convergence Default Optimize, on page 133

Feature Information for Fast Convergence Default Optimize

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 10: Feature Information for Fast Convergence Default Optimize

Feature Name	Releases	Feature Information
Fast Convergence Default Optimize	Cisco IOS XE Amsterdam 17.3.2	The fast convergence default optimize feature modifies the default settings of all the protocols to recommended defaults for fast convergence. No new commands were added or modified.

Information About Fast Convergence Default Optimize

The fast convergence default optimize feature modifies the default settings of all the protocols to recommended defaults for fast convergence. To revert the defaults to pre-fast-convergence settings for both IS-IS and OSPF, **no routing-default-optimize** command is used. This command sends signals to IS-IS and OSPF and modifies the default configuration for these protocols.

By default, the fast convergence settings is enabled which means when you upgrade the software, you can automatically see the new behavior. This makes easier integration of the devices in a multi-vendor deployment and reduces support cases for poor convergence.

When default optimize is disabled, existing protocol default configuration is used. When default optimize is enabled, new protocol defaults are used. The show running configurations does not display configuration lines for default settings even when default settings are being used.

A configuration of a protocol overrides the default, but a change to default optimize does not override any configuration.

The following is the sample output of **spf-interval** command in IS-IS:

```
Device(config-if)# router isis
Device(config-router)# spf-interval 10 5500 5500
```

If a non-default value is configured, it will be displayed in show running configuration output:

```
Device(config-router)# spf-interval 5 50 200
Device(config-router)# do show run | inc spf-interval
 spf-interval 5 50 200
```

You can revert to the default values by configuring the default values or by removing the non-default configuration.

Default Optimize Values for IS-IS

The following table summarizes the configuration impacted by default optimize:

IS-IS command	Parameters	Default optimize disabled	Default optimize enabled
fast-flood			
	# of lsps flooded back-back	Disabled	10
spf-interval			
	Initial (milliseconds)	5500	50
	Secondary (milliseconds)	5500	200
	max (seconds)	10	5
prc-interval			
	Initial (milliseconds)	2000	50
	Secondary (milliseconds)	5000	200
	max (seconds)	5	5
lsp-gen-interval			
	Initial (milliseconds)	50	50
	Secondary (milliseconds)	5000	200
	max (seconds)	5	5

IS-IS command	Parameters	Default optimize disabled	Default optimize enabled
log-adjacency-changes		disabled	enabled

Default Optimize Values for OSPF

The following table summarizes the configuration impacted by default optimize for OSPFv2/v3:

OSPF command	Parameters	Default optimize disabled	Default optimize enabled
timers throttle spf			
	Initial (milliseconds)	5000	50
	Secondary (milliseconds)	10000	200
	max (milliseconds)	10	5
timers throttle lsa all			
	Initial (milliseconds)	0	50
	Secondary (milliseconds)	5000	200
	max (milliseconds)	5	5
timers lsa arrival			
	milliseconds	1000	100

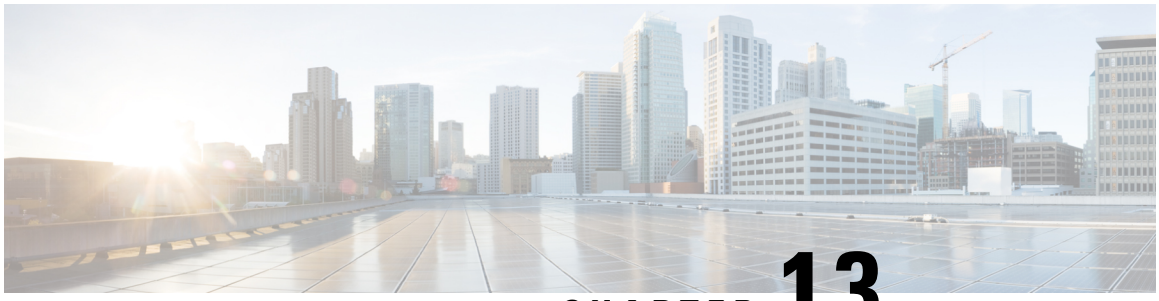
The following is the sample output of **show ip ospf** command for OSPFv2 with the default-optimize values.

```
Device# show ip ospf
 Routing Process "ospf 10" with ID 10.1.1.1
 Start time: 00:00:01.471, Time elapsed: 03:00:34.706
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 Supports NSSA (compatible with RFC 3101)
 Supports Database Exchange Summary List Optimization (RFC 5243)
 Event-log enabled, Maximum number of events: 1000, Mode: cyclic
 Router is not originating router-LSAs with maximum metric
 Initial SPF schedule delay 50 msecs
 Minimum hold time between two consecutive SPFs 200 msecs
 Maximum wait time between two consecutive SPFs 5000 msecs
 Incremental-SPF disabled
 Initial LSA throttle delay 50 msecs
 Minimum hold time for LSA throttle 200 msecs
 Maximum wait time for LSA throttle 5000 msecs
Minimum LSA arrival 100 msecs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
EXCHANGE/LOADING adjacency limit: initial 300, process maximum 300
Number of external LSA 18. Checksum Sum 0x075EB2
Number of opaque AS LSA 0. Checksum Sum 0x000000
```

```
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Number of areas transit capable is 0
External flood list length 0
IETF NSF helper support enabled
Cisco NSF helper support enabled
Reference bandwidth unit is 100 mbps
    Area BACKBONE(0)
        Number of interfaces in this area is 4 (2 loopback)
        Area has RRR enabled
        Area has no authentication
        SPF algorithm last executed 02:27:23.736 ago
        SPF algorithm executed 20 times
        Area ranges are
        Number of LSA 94. Checksum Sum 0x321DCF
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

The following is the sample output of **show ospf** command for OSPFv3 with the default-optimize values.

```
Device# show ospfv3
 OSPFv3 10 address-family ipv6
 Router ID 10.11.11.11
 Supports NSSA (compatible with RFC 3101)
Supports Database Exchange Summary List Optimization (RFC 5243)
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
Router is not originating router-LSAs with maximum metric
Initial SPF schedule delay 50 msecs
Minimum hold time between two consecutive SPFs 200 msecs
Maximum wait time between two consecutive SPFs 5000 msecs
Initial LSA throttle delay 50 msecs
Minimum hold time for LSA throttle 200 msecs
Maximum wait time for LSA throttle 5000 msecs
Minimum LSA arrival 100 msecs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Retransmission limit dc 24 non-dc 24
EXCHANGE/LOADING adjacency limit: initial 300, process maximum 300
Number of external LSA 0. Checksum Sum 0x000000
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Graceful restart helper support enabled
Reference bandwidth unit is 100 mbps
RFC1583 compatibility enabled
    Area BACKBONE(0)
        Number of interfaces in this area is 2
        SPF algorithm executed 7 times
        Number of LSA 3. Checksum Sum 0x012426
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```



CHAPTER 13

Routing Information Base Support

The Routing Information Base (RIB) enhancement supports route redistribution and on-demand nexthop requirements.

- Feature Information for Routing Information Base Support, on page 137
- Routing Information Base Support for Route Redistribution, on page 138
- OSPF Node SID Redistribution Support , on page 138
- Routing Information Base Support for On-Demand Next Hop, on page 140

Feature Information for Routing Information Base Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 11: Feature Information for Routing Information Base Support

Feature Name	Releases	Feature Information
Routing Information Base Support	Cisco IOS XE Amsterdam 17.3.2	The Routing Information Base (RIB) enhancement supports route redistribution and On-Demand Nexthop requirements. No new commands were added or modified.
OSPF Node SID Redistribution Support	Cisco IOS XE Amsterdam 17.3.2	When OSPF receives the redistributed prefixes from other IGPs and vice versa the prefix segment identifiers (SIDs) are also advertised which was not the case earlier. You need to have the BGP LS (or) segment routing mapping server (SRMS) support to learn the SIDs across the IGP domains. The following commands were added or modified for this feature: **show ip ospf rib redistribution detail**, **show ip ospf segment-routing local-prefix**, **show ip ospf segment-routing sid-database**, **show ip route 3.3.3.3**.

Routing Information Base Support for Route Redistribution

Effective with Cisco IOS XE Everest 16.5.1, a requirement to redistribute labels associated with prefixes is introduced. To support redistribution requirements, the storage of local label per prefix is supported in RIB.

The local label is stored instead of the SID to ease use with different protocols which may use different SRGBs. The SID assigned by the destination protocol may not be the same as the SID associated with the source protocol.

The prefix reachability advertisement or an SRMS advertisement is the source of the SID. In SRMS advertisement, the destination protocols for redistribution do not advertise the SID in their prefix reachability advertisements, as this alters conflict resolution by indicating on other network nodes that the source of the advertisement was not from SRMS.

OSPF Node SID Redistribution Support

Effective Cisco IOS XE 16.7.1, when OSPF receives the redistributed prefixes from other IGPs and vice versa the prefix segment identifiers (SIDs) are also advertised which was not the case earlier. You needed to have the BGP LS (or) segment routing mapping server (SRMS) support to learn the SIDs across the IGP domains.

When the user enable redistribution under OSPF the prefix SID entries associated with the prefix entries are provided to OSPF. This gets advertised by OSPF to all its neighbor. The way OSPF advertises varies depending upon the role of OSPF in the network.

Information About OSPF Node SID Redistribution Support

NSSA ASBR

When you enable **redistribute ISIS** *instance* **ip** under OSPF which is Not-So-Stubby Area autonomous system boundary router (NSSA ASBR), it gets all the prefixes from IP routing information base (RIB) which are learnt by IS-IS along with the SID entries. OSPF generates Extended Prefix LSA (EPL) with the scope as area and the route type as RTYPE_NSSA1 or RTYPE_NSSA2 for the prefixes and advertises to all its neighbors. Similarly, when the redistribution is un-configured (or) when the prefixes become unavailable OSPF withdraws the EPL. When the redistributed route is a non-connected route then the OSPF sets the No-PHP flag but explicit NULL flag is not set. However, when the redistributed route is a connected route then OSPF sets the explicit NULL and No-PHP flag according to the configuration done in the SR policy.

When NSSA ABR receives the EPL, the ABR translates the LSA into opaque AS EPL and floods it to all its neighbors.

When a NSSA router which is neither ABR nor ASBR receives the EPL, it learns the prefix along with the SID entries and floods it to all its neighbors in the same area.

non-NSSA ASBR

When the user enabled **redistribute ISIS** *instance* **ip** under OSPF which is regular ASBR router, it gets all the prefixes from IP RIB which are learnt by IS-IS along with the SID entries. OSPF generates EPL with the scope as autonomous system (AS) and the route type as RTYPE_EXTERN1 or RTYPE_EXTERN2 for the prefixes and advertises to all its neighbors. Similarly when the redistribution is unconfigured (or) when the prefixes become unavailable, OSPF withdraws the EPL again with AS-Scope. When the redistributed route

is a non-connected route then the OSPF sets the No-PHP flag but explicit NULL flag is not set. However, when the redistributed route is a connected route then OSPF sets the explicit NULL and No-PHP flag according to the configuration done in the SR policy. When a router receives the EPL with AS scope, it learns the prefix along with the SID entry and floods it to all its neighbors in all areas.

Redistributing Prefix

When IS-IS is enabled for redistribution of OSPF routes the prefixes are given along with the SID information so that the prefixes reach to other domain with the SID values. Refer to the below topology to understand the OSPF prefixes redistribution to the other domains:

Figure 17: OSPF Prefix Redistribution



R1 and R2 are enabled for OSPF. R2 and R3 are enabled for IS-IS. Both IS-IS and OSPF are enabled for Segment Routing. In R2, both IS-IS and OSPF are configured. Prefixes configured are:

1. 10.1.1/32 in R1 (enabled for OSPF with SID 1)
2. 10.2.2/32 in R2 (enabled for OSPF with SID 2)
3. 10.3.3/32 in R3 (enabled for ISIS SID 3)

When you enable SID redistribution in R2, then the prefix 10.3.3.3/32is redistributed to R1. So, R1 knows the SID to reach the prefix R3.

```
conf trouter isis 10 net 49.0001.0000.0000.0001.00 metric-style wide distribute link-state
 segment-routing mpls router ospf 10 router-id 10.2.2.2 segment-routing mpls distribute
link-state
```

To enable redistribution of ISIS into OSPF routes:

```
conf t router ospf 10 redistribute isis 10 ip
```

Verify OSPF Node SID Redistribution

Use the **show ip ospf rib redistribution detail** command to verify if OSPF is redistributing the prefixes from IS-IS.

Note C8xxx=C8200/C8300/C8500 or C8000v

```
c8xxx# show ip ospf rib redistribution  detail
OSPF Router with ID (10.2.2.2) (Process ID 10)

               Base Topology (MTID 0)

OSPF Redistribution
10.3.3.3/32, type 2, metric 20, tag 0, from IS-IS Router
  Attributes 0x1000000, event 1, PDB Index 4, PDB Mask 0x0
  Source route metric 20, tag 0
  SID 1003, SID Flags NP-bit, EPX Flags None
   via 10.9.0.9, Ethernet0/0
```

Use the **show ip ospf segment-routing local-prefix** command to verify if the SID entries are advertised to its neighbor.

```
c8xxx# show ip ospf segment-routing local-prefix

            OSPF Router with ID (10.2.2.2) (Process ID 10)
Area 0:
 Prefix:            Sid:   Index:          Type:      Source:
10.2.2.2/32         2      10.0.0.0        Intra      Loopback0
AS external:
Prefix:            Sid:   Index:          Type:      Source:
10.3.3.3/32        3      10.0.0.1        External   Redist
```

Use the **show ip ospf segment-routing sid-database** command to verify if the SIDs are received.

```
Device# show ip ospf segment-routing sid-database

            OSPF Router with ID (10.1.1.1) (Process ID 10)
OSPF Segment Routing SIDs

Codes: L - local, N - label not programmed,
       M - mapping-server

SID             Prefix              Adv-Rtr-Id       Area-Id  Type
--------------  ------------------  ---------------  -------  ----------
1               10.1.1.1/32          10.1.1.1         0        Intra
2               10.2.2.2/32          10.2.2.2         0        Intra
3               10.3.3.3/32          10.2.2.2         -        External
```

Use the **show ip route 10.3.3.3** command to verify if the IP routing entry is configured for the redistributed route.

```
c8xxx# show ip route 10.3.3.3
Routing entry for 10.3.3.3/32
  Known via "ospf 10", distance 110, metric 20, type extern 2, forward metric 20
  Last update from 10.2.0.2 on Ethernet0/1, 00:00:01 ago
 SR Incoming Label: 16003
  Routing Descriptor Blocks:
  * 10.3.1.3, from 10.2.2.2, 00:00:01 ago, via Ethernet1/1, merge-labels
      Route metric is 20, traffic share count is 1
      MPLS label: 16003
      MPLS Flags: NSF
```

Routing Information Base Support for On-Demand Next Hop

For On-Demand Next Hop (ODN) requirements, RIB supports a next hop called binding label which is provided by the supporting routing protocol (BGP). The binding label is used by the FIB to dynamically resolve the next hop.

The route producer installs a local binding label which identifies the ODN tunnel path associated with the next hop. The labeled traffic is sent via the tunnel and the label is distinct from the existing outlabel.

The following is the sample output of **show ip route** command where each next hop is updated to show the binding label.

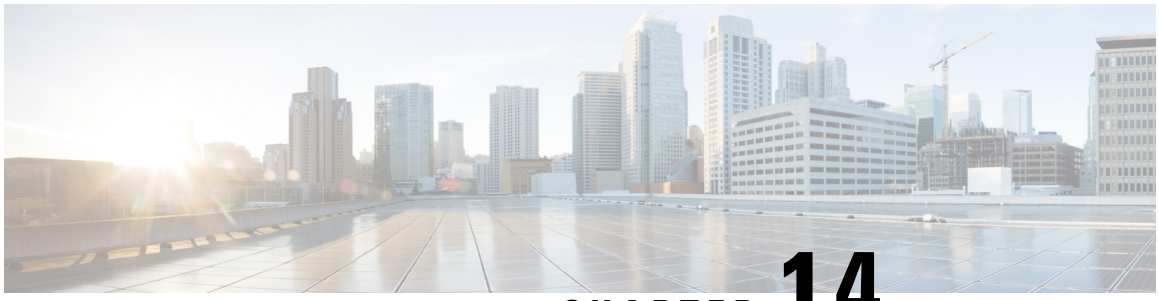
```
Device# show ip route 10.10.10.2

Routing entry for 10.10.10.2/32
```

```
Known via "isis", distance 115, metric 10, type level-1
Redistributing via isis
Last update from 10.200.200.2 on Ethernet0/0, 00:00:14 ago
Incoming Label: 16100
Routing Descriptor Blocks:
* 10.200.200.2, from 10.10.10.2, 00:00:14 ago, via Ethernet0/0
    Route metric is 10, traffic share count is 1
    * Binding Label 4020, from 10.2.2.2, 00:00:14 ago,
    Route metric is 10, traffic share count is 1
```



Note The incoming labels are seen only after the SID redistribution is enabled.



CHAPTER 14

SR-TE On Demand LSP

The SR TE On demand LSP feature provides the ability to connect Metro access rings via a static route to the destination. The static route is mapped to an explicit path and that will trigger an on demand LSP to the destination. The SR TE On demand LSP feature will be used to transport the VPN services between the Metro access rings.

- Feature Information for SR-TE On Demand LSP, on page 143
- Restrictions for SR-TE On Demand LSP, on page 144
- Information About SR-TE On Demand LSP, on page 144
- How to Configure SR-TE On Demand LSP, on page 145
- **Configure Native UCMP for Static Routing**, on page 148

Feature Information for SR-TE On Demand LSP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 12: Feature Information for SR-TE On Demand LSP

Feature Name	Releases	Feature Information
SR-TE On Demand LSP	Cisco IOS XE Amsterdam 17.3.2	The SR TE On demand LSP feature provides the ability to connect Metro access rings via a static route to the destination. The static route is mapped to an explicit path and that will trigger an on demand LSP to the destination. The SR TE On demand LSP feature will be used to transport the VPN services between the Metro access rings. The following command was modified: **mpls traffic-eng auto-tunnel**.

Restrictions for SR-TE On Demand LSP

- Segment-Routing auto tunnel static route does not support ECMP.
- Metrics for IP explicit path and administrtive distance change for auto tunnel SRTE static route is not supported.
- MPLS Traffic Engineering (TE) Nonstop Routing (NSR) must be configured on the active route processor (RP) for Stateful Switchover (SSO). This is because, SR static auto tunnel will fail to come up after SSO, unless the static route auto tunnel configuration is removed and reconfigured.
- IP unnumbered interfaces do not support dynamic path.
- When using IP unnumbered interfaces, you cannot specify next hop address as an explicit path index. It should be a node address or a label.

Information About SR-TE On Demand LSP

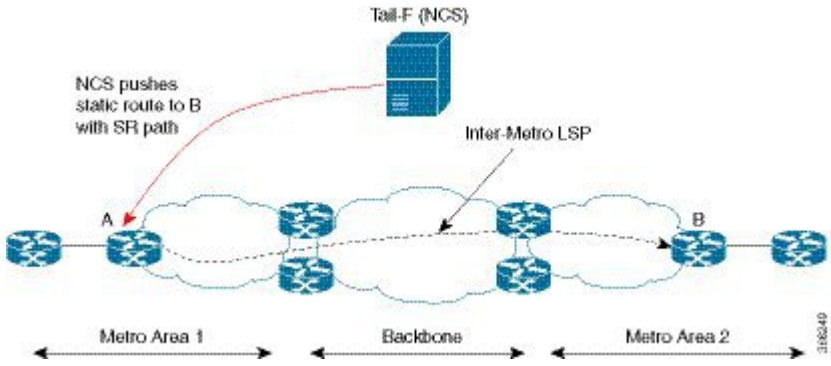
The SR TE On demand LSP feature provides the ability to connect Metro access rings via a static route to the destination.

SR-TE: Setup LSP as Static Route

Agile Carrier Ethernet (ACE) solution leverages Segment Routing-based transport for consolidated VPN services. In metro rings architecture, the access rings do not share their routing topologies with each other.

The SR TE On demand LSP feature provides the ability to connect Metro access rings via a static route to the destination. The static route is mapped to an explicit path and that will trigger an on demand LSP to the destination. The SR TE On demand LSP feature will be used to transport the VPN services between the Metro access rings.

Figure 18: Inter-Metro LSP in ACE Solution



Inter-Metro LSPs have the following aspects:

- The source packet may not know the IP address of the destination device.
- Existing segment routing features are applicable for LSPs.

The binding SID helps in steering the traffic in the SR-TE tunnel. In other words, ingress MPLS packet with the binding SID will be forwarded through the specific SR-TE tunnel.

Static SRTE over Unnumbered Interfaces

As explained in the previous section, you can set up LSP as static route to create an auto tunnel by specifying an IP explicit path.

The explicit path is a combination of IP addresses (or) IP address and labels. You can also configure the static SRTE tunnel over unnumbered interfaces. There are few restrictions for unnumbered interfaces against numbered interfaces.

- You must specify the node IP address, not the next hop interface address in the ip-explicit path option.
- You must not specify adjacency SID in the explicit path option. In short, the explicit path option should contain only the node IP address (/32 mask) and prefix SID labels.

How to Configure SR-TE On Demand LSP

Perform the following steps to configure SR-TE On Demand LSP.

Configuring LSP as Static Route

To avoid packet drop after RP switchover with SR TE, it is recommended to use the following command:

```
mpls traffic-eng nsr
```

If ISIS is configured, use the following command:

```
router isis
 nsf cisco
 nsf interval 0
```

Enabling Segment Routing Auto Tunnel Static Route

Perform this task to configure auto tunnel static route as follows:

- Configure IP explicit path
- Associate the auto tunnel with an IP explicit path with a static route
- Enable peer-to-peer (P2P) auto tunnel service

```
ip explicit-path name path1
 index 1 next-label 16002
 index 2 next-label 16006
 exit
ip route 172.16.0.1 255.240.0.0 segment-routing mpls path name path1
mpls traffic-eng auto-tunnel p2p
mpls traffic-eng auto-tunnel p2p config unnumbered-interface loopback0
mpls traffic-eng auto-tunnel p2p tunnel-num min 10 max 100
```

Verifying Segment Routing Auto-Tunnel Static Route

The command **show mpls traffic-eng service summary** displays all registered TE service clients and statistics that use TE auto tunnel.

```
Device# show mpls traffic-eng service summary

Service Clients Summary:
  Client: BGP TE
    Client ID                 :0
    Total P2P tunnels         :1
    P2P add requests          :6
    P2P delete requests       :5
    P2P add falis             :0
    P2P delete falis          :0
    P2P notify falis          :0
    P2P notify succs          :12
    P2P replays               :0
  Client: ipv4static
    Client ID                 :1
    Total P2P tunnels         :1
    P2P add requests          :6
    P2P delete requests       :5
    P2P add falis             :0
    P2P delete falis          :0
    P2P notify falis          :0
    P2P notify succs          :85
    P2P replays               :0
```

The command **show mpls traffic-eng auto-tunnel p2p** displays the peer-to-peer (P2P) auto tunnel configuration and operation status.

```
Device# show mpls traffic-eng auto-tunnel p2p

State: Enabled
  p2p auto-tunnels: 2 (up: 2, down: 0)
  Default Tunnel ID Range: 62336 - 64335
  Config:
   unnumbered-interface: Loopback0
   Tunnel ID range: 1000 - 2000
```

The command **show mpls traffic-eng tunnel summary** displays the status of P2P auto tunnel.

```
Device# show mpls traffic-eng tunnel summmary

Signalling Summary:
    LSP Tunnels Process:            running
    Passive LSP Listener:           running
    RSVP Process:                   running
    Forwarding:                     enabled
    auto-tunnel:
        p2p    Enabled  (1), id-range:1000-2000
    Periodic reoptimization:        every 3600 seconds, next in 1265 seconds
    Periodic FRR Promotion:         Not Running
    Periodic auto-bw collection:    every 300 seconds, next in 66 seconds
    SR tunnel max label push:       13 labels
    P2P:
      Head: 11 interfaces,   5234 active signalling attempts, 1 established
            5440 activations,  206 deactivations
            1821 failed activations
            0 SSO recovery attempts, 0 SSO recovered
      Midpoints: 0, Tails: 0
```

```
    P2MP:
      Head: 0 interfaces,   0 active signalling attempts, 0 established
            0 sub-LSP activations,  0 sub-LSP deactivations
            0 LSP successful activations,  0 LSP deactivations
            0 SSO recovery attempts, LSP recovered: 0 full, 0 partial, 0 fail
      Midpoints: 0, Tails: 0
Bidirectional Tunnel Summary:
    Tunnel Head: 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

AutoTunnel P2P Summary:
    ipv4static:
        Tunnels: 1 created, 1 up, 0 down
    Total:
        Tunnels: 1 created, 1 up, 0 down
```

The command **show mpls traffic-eng tunnel auto-tunnel** only displays TE service auto tunnel.

```
Device# show mpls traffic-eng tunnel auto-tunnel detail

 P2P TUNNELS/LSPs:

Name: R1_t1000                            (Tunnel1000) Destination: 10.0.0.0 Ifhandle: 0x17
 (auto-tunnel for ipv4static)
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, (SEGMENT-ROUTING) type explicit (verbatim) path202 (Basis for Setup)

  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    Path Selection:
     Protection: any (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
    Hop Limit: disabled [ignore: Verbatim Path Option]
    Cost Limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear
    AutoRoute: disabled LockDown: disabled Loadshare: 0 [0] bw-based
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: enabled

  History:
    Tunnel:
      Time since created: 33 days, 20 hours, 29 minutes
      Time since path change: 10 days, 19 hours, 45 minutes
      Number of LSP IDs (Tun_Instances) used: 1646
    Current LSP: [ID: 1646]
      Uptime: 10 days, 19 hours, 45 minutes
    Prior LSP: [ID: 1645]
      ID: path option unknown
      Removal Trigger: signalling shutdown
  Tun_Instance: 1646
  Segment-Routing Path Info (IGP information is not used)
    Segment0[First Hop]: 10.0.0.0, Label: 16002
    Segment1[ - ]: Label: 16006
```

The command **show mpls traffic-eng tunnel brief** displays auto tunnel information.

```
Device# show mpls traffic-eng tunnel brief

Signalling Summary:
    LSP Tunnels Process:            running
    Passive LSP Listener:           running
    RSVP Process:                   running
    Forwarding:                     enabled
    auto-tunnel:
        p2p    Enabled  (2), id-range:1000-2000

    Periodic reoptimization:        every 3600 seconds, next in 406 seconds
    Periodic FRR Promotion:         Not Running
    Periodic auto-bw collection:    every 300 seconds, next in 107 seconds
    SR tunnel max label push:       13 labels

P2P TUNNELS/LSPs:
TUNNEL NAME                     DESTINATION      UP IF     DOWN IF   STATE/PROT
R1_t1                           10.66.66.66      -         -         up/down
R1_t2                           10.66.66.66      -         -         up/up
R1_t3                           10.66.66.66      -         -         up/up
R1_t10                          10.66.66.66      -         -         up/up
SBFD tunnel                     10.33.33.33      -         -         up/up
SBFD Session configured: 1      SBFD sessions UP: 1
```

Configure Native UCMP for Static Routing

In a network where traffic is load balanced on two or more links, configuring equal metrics on the links would create Equal Cost Multipath (ECMP) next hops. Because the bandwidth of the links is not taken into consideration while load balancing, the higher bandwidth links are underutilized. To avoid this problem, you can configure Unequal Cost Multipath (UCMP), either locally (local UCMP), or natively (native UCMP) so that the higher bandwidth links carry traffic in proportion to the capacity of the links. UCMP supports IPv4 and IPv6 static VRF routes.

Local UCMP

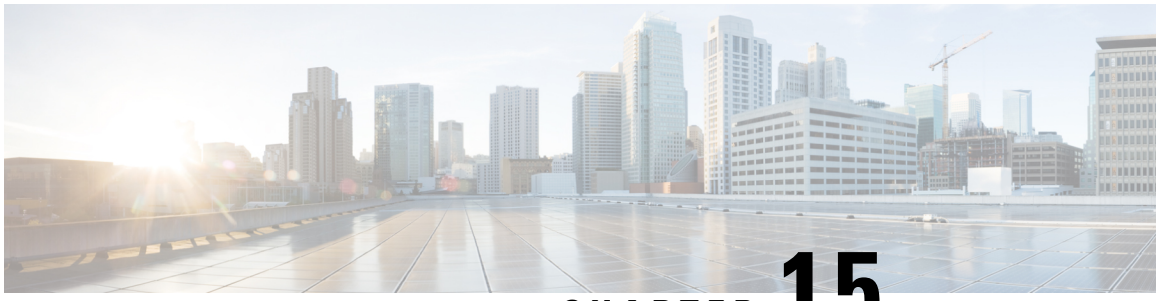
All static routes are configured with the same link metrics. The static IGP calculates the load metric based on the bandwidth of the links and load balances the traffic across the links. However, local UCMP does not consider bandwidth while load balancing across links that are closer to the destination (multiple hops away).

Native UCMP

Static routes over higher bandwidth links are configured with lower link metrics so that they are preferred to routes over lower bandwidth links. The static IGP calculates the load metric based on the bandwidth of the links and determines the percentage of traffic going out of the higher and lower bandwidth links. By matching the configured link metrics with end-to-end available bandwidth, native UCMP is able to effectively load balance traffic across links that are closer to the destination (multiple hops away).

Configuration Example

Consider the topology in the following figure. For load balancing traffic out of Router A1, if local UCMP is used, then both 10G and 100G links will have equal link metrics. The static IGP decides to send more traffic out of the 100G link because of the higher load metric. However, for load balancing traffic out of Router A2, local UCMP works only on links to Routers C1 and C2. For load balancing traffic from Router C1 to Router A1 and Router C2 to Router A1, native UCMP is preferred. As a result, local UCMP is used only on single hop destinations, and native UCMP is used for multi-hop destinations.



CHAPTER 15

Segment Routing MPLS OAM Support

Segment Routing Operations, Administration, and Maintenance (OAM) helps service providers to monitor label-switched paths (LSPs) and quickly isolate forwarding problems to assist with fault detection and troubleshooting in the network. The Segment Routing OAM feature provides support for Nil-FEC (forwarding equivalence classes) LSP Ping and Traceroute, IGP prefix SID FEC type, and partially IGP adjacency-SID FEC type for SR-TE functionality.

- Feature Information for Segment Routing OAM Support, on page 151
- Restrictions for Segment Routing OAM MPLS Support, on page 152
- Information About Segment Routing MPLS OAM Support, on page 152
- How to Diagnose Segment Routing with LSP Ping and Trace Route Nil FEC Target, on page 154
- Example for LSP Ping Nil FEC Target Support, on page 154
- Path Validation in Segment Routing Network, on page 156
- Configuring Segment Routing MPLS Traffic Engineering for MPLS Ping and Traceroute, on page 158
- Configuring Segment Routing MPLS IGP for MPLS Ping and Traceroute, on page 159

Feature Information for Segment Routing OAM Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 13: Feature Information for Segment Routing OAM Support

Feature Name	Releases	Feature Information
Segment Routing OAM Support	Cisco IOS XE Amsterdam 17.3.2	The Segment Routing OAM feature provides support for Nil-FEC (forwarding equivalence classes) LSP Ping and Traceroute functionality. The Nil-FEC LSP ping and traceroute operation are simply extension of regular MPLS ping and trace route.

Restrictions for Segment Routing OAM MPLS Support

- Ping and traceroute are unsupported with SR-TE static auto tunnel, BGP Dynamic TE, and on-demand next hop auto tunnels.
- Strict-SID option is not supported by the path installed by OSPF.
- MPLS traceroute does not support popping of two explicit null labels in one node.
- Rerouting the path to IP over MPLS segment without using Layer3 VPN is not supported due to IP routing destination not being a MPLS FEC.

Information About Segment Routing MPLS OAM Support

Segment Routing OAM Support

The Nil-FEC LSP ping and traceroute operations are extensions of regular MPLS ping and traceroute . Nil-FEC LSP Ping/Trace functionality support Segment Routing and MPLS Static. It also act as an additional diagnostic tool for all other LSP types. This feature allows operators to test any label stack to specify the following:

- label stack
- outgoing interface
- nexthop address

In the case of segment routing, each segment nodal label and adjacent label along the routing path is put into the label stack of an echo request message from initiator Label Switch Router (LSR); MPLS data plane forward this packet to the label stack target, and the label stack target reply the echo message back.



Note On IOS XE platforms, **traceroute mpls ipv4** <*prefix*> uses LDP labels by default, even if **sr-label-preferred** command is configured. To use SR labels, specify the FEC type in the command. This behavior is the same on all IOS XE platforms.

Benefits of Segment Routing OAM Support

- The feature enables the MPLS OAM functionality in the Segment Routing Network where the traffic is engineering via SR-TE tunnels or native SR forwarding.
- In traditional MPLS networks, source node chooses the path based on hop by hop signaling protocols such as LDP or RSVP-TE. In Segment Routing Networks, the path is specified by set of segments which are advertised by the IGP protocols (currently OSPF and ISIS).
- As the volume of services offered using SR increase, it is important that the operator essentially is able to do the connectivity verification and the fault isolation in the SR architecture.

- The segment assignment is not based on hop by hop protocols as in traditional MPLS network, any broken transit node could lead to null routes, which could lead to undesired trraffic behavior.
- Both SR and SR-TE supports load balancing, it is important to trace all the ECMP paths available between source and target routers. The features offers the multipath traceroute support for both TE and native SR paths.
- The following are the main benefits of Segment Routing-OAM Support:
  - **Operations**: Network monitoring and fault management.
  - **Administration**: Network discovery and planning.
  - **Maintenance**: Corrective and preventive activities, minimize occurrences and impact of failures.

Segment Routing MPLS Ping

MPLS ping and traceroute are extendable by design. You can add SR support by defining new FECs and/or additional verification procedures. MPLS ping verifies MPLS data path and performs the following:

- Encapsulates echo request packet in MPLS labels.
- Measures coarse round trip time.
- Measures coarse round trip delay.

Segment Routing MPLS Traceroute

MPLS ping and traceroute are extendable by design. You can add SR support by defining new forwarding equivalence classes (FECs) and/or additional verification procedures. MPLS traceroute verifies forwarding and control plane at each hop of the LSP to isolate faults. Traceroute sends MPLS echo requests with monotonically increasing time-to-live (TTL), starting with TTL of 1. Upon TTL expiry, transit node processes the request in software and verifies if it has an LSP to the target FEC and intended transit node. The transit node sends echo reply containing return code specifying the result of above verification and label stack to reach the next-hop, as well as ID of the next-hop towards destination, if verification is successful. Originator processes echo reply to build the next echo request containing TTL+1. Process is repeated until the destination replies that it is the egress for the FEC.

LSP Ping Operation for Nil FEC target

The LSP Ping/Traceroute is used in identifying LSP breakages. The nil-fec target type can be used to test the connectivity for a known label stack. Follow the existing LSP ping procedure (for more information, refer MPLS LSP Ping/Traceroute), with the following modifications:

- Build the echo request packet with the given label stack.
- Append explicit null label at the bottom of the label stack.
- Build echo request FTS TLV with target FEC Nil FEC and label value set to the bottom label of the label stack, which is explicit-null.

How to Diagnose Segment Routing with LSP Ping and Trace Route Nil FEC Target

Using LSP Ping for Nil FEC Target

The Nil FEC LSP ping and traceroute operation are simply extension of regular MPLS ping and trace route. **nil-fec labels <label, label…>** is added to the ping mpls command. This command sends an echo request message with MPLS label stack as specified and add another explicit null at bottom of the stack.

```
ping mpls nil-fec labels <comma separated labels> output interface <tx-interface> nexthop
<nexthop ip addr>
[repeat <count>]
[size <size> | sweep <min_size> <max_size> <increment>]
[timeout <seconds>]
[interval <milliseconds>]
[destination <addr_start> [<addr_end> [<addr_incr_mask> | <addr_incr>]]]
[source <addr>]
[exp <exp-value>]
[pad <pattern>]
[ttl <ttl>]
[reply [mode [ipv4 | router-alert | no-reply]]
[dscp <dscp-bits>]
[pad-tlv]]
[verbose]
[force-disposition ra-label]
{dsmap | ddmap [l2ecmp]} [hashkey {none | {ipv4 | ipv4-label-set {bitmap <bitmap_size>}}]
```

For more information, refer ping mpls.

Using LSP Traceroute for Nil FEC Target

```
trace mpls nil-fec labels <comma separated labels> output interface <tx-interface>} nexthop
 <nexthop ip addr>
[timeout <seconds>]
[destination <addr_start> [<addr_end> [<addr_incr_mask> | <addr_incr>]]]
[source <addr> ]
[exp <exp-value>]
[ttl <ttl-max>]
[reply [mode [ipv4 | router-alert | no-reply]]
[dscp <dscp-bits>]
[pad-tlv]]
[flags {fec | ttl}]
[hashkey ipv4 | ipv4-label-set {bitmap <bitmap_size>}]
```

For more information, refer to the traceroute mpls.

Example for LSP Ping Nil FEC Target Support

```
Node loopback IP address: 10.1.1.3              10.1.1.4              10.1.1.5
      1.1.1.7
Node label:                                      16004                 16005
```

```
     16007
Nodes:                    Arizona ---------------- Utah  ---------------- Wyoming
--------------- Texas
Interface:                      Eth1/0        Eth1/0
Interface IP address:          10.30.1.3      10.30.1.4

Device#sh mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         Pop Label  3333.3333.0000-Et1/0-10.30.1.3   \
                                       0             Et1/0      10.30.1.3
17         Pop Label  5555.5555.5555-Et1/1-10.90.1.5   \
                                       0             Et1/1      10.90.1.5
18         Pop Label  3333.3333.0253-Et0/2-102.102.102.2   \
                                       0             Et0/2      10.102.102.2
19         Pop Label  10.9.9.4/32      0             Et0/2      10.102.102.2
20         Pop Label  10.1.1.5/32      0             Et1/1      10.90.1.5
21         Pop Label  10.1.1.3/32      0             Et1/0      10.30.1.3
22         Pop Label  10.16.16.16/32   0             Et1/0      10.30.1.3
23         Pop Label  10.16.16.17/32   0             Et1/0      10.30.1.3
24         Pop Label  10.17.17.17/32   0             Et1/0      10.30.1.3
25         20         10.9.9.3/32      0             Et1/0      10.30.1.3
26         21         10.1.1.6/32      0             Et1/0      10.30.1.3
27         24         10.1.1.2/32      0             Et1/0      10.30.1.3
           28         10.1.1.2/32      0             Et1/1      10.90.1.5
28         18         10.1.1.7/32      0             Et1/1      10.90.1.5
29         27         10.9.9.7/32      0             Et1/1      10.90.1.5
30         Pop Label  10.55.1.0/24     0             Et1/1      10.90.1.5
31         Pop Label  10.19.1.0/24     0             Et1/0      10.30.1.3
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
32         Pop Label  10.1.1.0/24      0             Et1/0      10.30.1.3
33         Pop Label  10.100.100.0/24  0             Et1/0      10.30.1.3
34         Pop Label  10.1.1.0/24      0             Et1/0      10.30.1.3
35         28         10.1.1.0/24      0             Et1/0      10.30.1.3
36         29         10.101.101.0/24  0             Et1/0      10.30.1.3
37         29         10.65.1.0/24     0             Et1/1      10.90.1.5
38         33         10.104.104.0/24  0             Et1/0      10.30.1.3
           39         10.104.104.0/24  0             Et1/1      10.90.1.5
39         30         10.103.103.0/24  0             Et1/1      10.90.1.5
16005      Pop Label  10.1.1.5/32      1782          Et1/1      10.90.1.5
16006      16006      10.1.1.6/32      0             Et1/0      10.30.1.3
16007      16007      10.1.1.7/32      0             Et1/1      10.90.1.5
16017      16017      10.17.17.17/32   0             Et1/0      10.30.1.3
16250      16250      10.9.9.3/32      0             Et1/0      10.30.1.3
16252      16252      10.9.9.7/32      0             Et1/1      10.90.1.5
16253      Pop Label  10.9.9.4/32      0             Et0/2      10.102.102.2
17000      17000      10.16.16.16/32   0             Et1/0      10.30.1.3
17002      17002      10.1.1.2/32      0             Et1/0      10.30.1.3
           17002      10.1.1.2/32      0             Et1/1      10.90.1.5

Device#ping mpls nil-fec labels 16005,16007 output interface ethernet 1/0 nexthop 10.30.1.4
 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,
  'l' - Label switched with FEC change, '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


Device#traceroute mpls nil-fec labels 16005,16007 output interface ethernet 1/0 nexthop
10.30.1.4
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,
  'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 10.30.1.3 MRU 1500 [Labels: 16005/16007/explicit-null Exp: 0/0/0]
L 1 10.30.1.4 MRU 1500 [Labels: implicit-null/16007/explicit-null Exp: 0/0/0] 1 ms
L 2 10.90.1.5 MRU 1500 [Labels: implicit-null/explicit-null Exp: 0/0] 1 ms
! 3 10.55.1.7 1 ms
```

Path Validation in Segment Routing Network

The MPLS OAM mechanisms help with fault detection and isolation for a MPLS data-plane path by the use of various target FEC stack sub-TLVs that are carried in MPLS echo request packets and used by the responder for FEC validation. While it is obvious that new sub-TLVs need to be assigned for segment routing, the unique nature of the segment routing architecture raises the need for additional operational considerations for path validation.

The forwarding semantic of Adjacency Segment ID is to pop the Segment ID and send the packet to a specific neighbor over a specific link. A malfunctioning node may forward packets using Adjacency Segment ID to an incorrect neighbor or over an incorrect link. The exposed Segment ID (of an incorrectly forwarded Adjacency Segment ID) might still allow such packet to reach the intended destination, although the intended strict traversal has been broken. MPLS traceroute may help with detecting such a deviation.

The format of the following Segment ID sub-TLVs follows the philosophy of Target FEC Stack TLV carrying FECs corresponding to each label in the label stack. This allows LSP ping/traceroute operations to function when Target FEC Stack TLV contains more FECs than received label stack at responder nodes. Three new sub-TLVs are defined for Target FEC Stack TLVs (Type 1), Reverse-Path Target FEC Stack TLV (Type 16) and Reply Path TLV (Type 21).

sub-Type	Value Field
34	IPv4 IGP-Prefix Segment ID
35	IPv6 IGP-Prefix Segment ID
36	IGP-Adjacency Segment ID

MPLS Ping and Traceroute for IGP Prefix-SID FEC Type

MPLS ping and traceroute operations for prefix SID are supported for various IGP scenarios, for example:

- Within an IS-IS level or OSPF area
- Across IS-IS levels or OSPF areas
- Route redistribution from IS-IS to OSPF and from OSPF to IS-IS

The MPLS LSP Ping feature is used to check the connectivity between ingress Label Switch Routers (LSRs) 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. The destination IP address of the MPLS echo request packet is different from the address used to select the label stack.

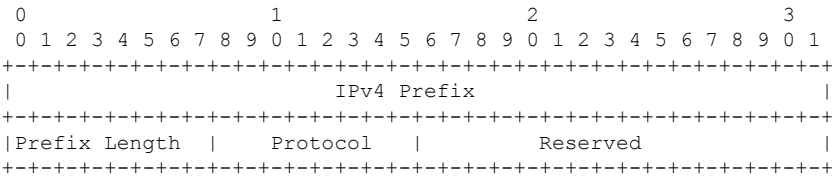
The MPLS LSP Traceroute feature is used to isolate the failure point of an LSP. It is used for hop-by-hop fault localization and path tracing. The MPLS LSP Traceroute feature relies on the expiration of the Time to Live (TTL) value of the packet that carries the echo request. When the MPLS echo request 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 MPLS LSP Tree Trace (traceroute multipath) operation is also supported for IGP Prefix SID. MPLS LSP Tree Trace provides the means to discover all possible equal-cost multipath (ECMP) routing paths of an LSP to reach a destination Prefix SID. It uses multipath data encoded in echo request packets to query for the load-balancing information that may allow the originator to exercise each ECMP. When the packet TTL expires at the responding node, the node returns the list of downstream paths, as well as the multipath information that can lead the operator to exercise each path in the MPLS echo reply. This operation is performed repeatedly for each hop of each path with increasing TTL values until all ECMP are discovered and validated.

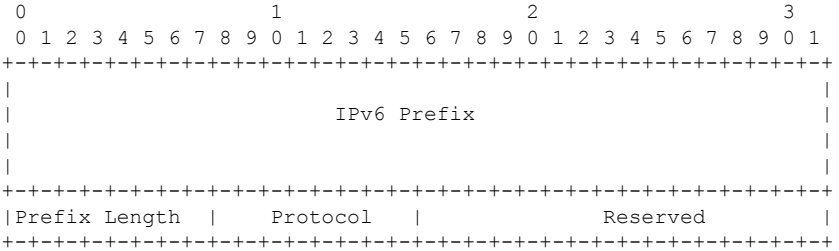
MPLS echo request packets carry Target FEC Stack sub-TLVs. The Target FEC sub-TLVs are used by the responder for FEC validation. The IGPIPv4 prefix sub-TLV has been added to the Target FEC Stack sub-TLV. The IGP IPv4 prefix sub-TLV contains the prefix SID, the prefix length, and the protocol (IS-IS or OSPF).

The network node which advertised the Node Segment ID is responsible for generating a FEC Stack Change sub-TLV with pop operation type for Node Segment ID, regardless of whether penultimate hop popping (PHP) is enabled or not.

The format is as below for IPv4 IGP-Prefix Segment ID:



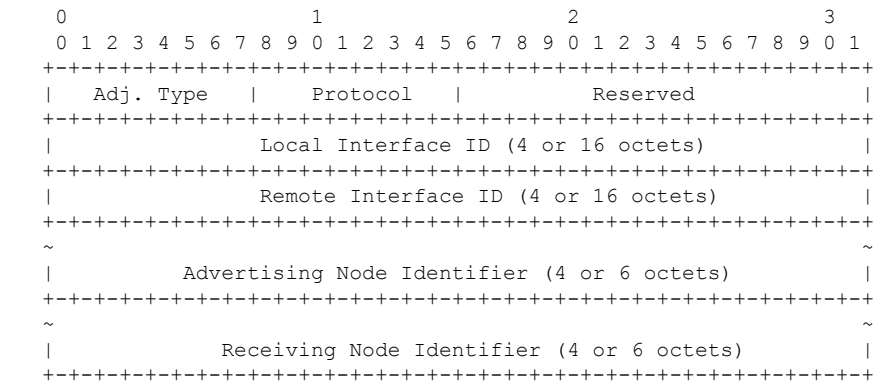
The format is as below for IPv6 IGP-Prefix Segment ID:



MPLS Ping and Traceroute for IGP-Adjacency Segment ID

The network node that is immediate downstream of the node which advertised the Adjacency Segment ID is responsible for generating FEC Stack Change sub-TLV for "POP" operation for Adjacency Segment ID.

The format is as below for IGP-adjacency SID:



Configuring Segment Routing MPLS Traffic Engineering for MPLS Ping and Traceroute

```
ping mpls traffic-eng tunnel <tun-id>
[repeat <count>]
[size <size> | sweep <min_size> <max_size> <increment>]
[timeout <seconds>]
[interval <milliseconds>]
[destination <addr_start> [<addr_end> [<addr_incr_mask> | <addr_incr>]]]
[source <addr>]
[exp <exp-value>]
[pad <pattern>]
[ttl <ttl>]
[reply [mode [ipv4 | router-alert | no-reply]]
[dscp <dscp-bits>]
[pad-tlv]]
[verbose]
[output {interface <tx-interface>} [nexthop <nexthop ip addr>]]
[{dsmap | ddmap [l2ecmp]} [hashkey {none | {ipv4 | ipv4-label-set {bitmap <bitmap_size>}}]

traceroute mpls [multipath] traffic-eng <tunnel-interface>
[timeout <seconds>]
[destination <addr_start> [<addr_end> [<addr_incr_mask> | <addr_incr>]]]
[source <addr> ]
[exp <exp-value>]
[ttl <ttl-max>]
[reply [mode [ipv4 | router-alert | no-reply]]
[pad-tlv]]
[output {interface <tx-interface>} [nexthop <nexthop ip addr>]]
[flags {fec | ttl}]
[hashkey ipv4 | ipv4-label-set {bitmap <bitmap_size>}]
```

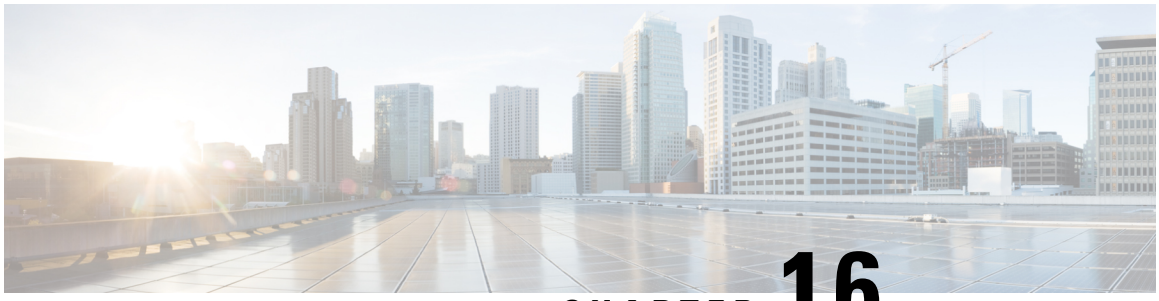
Configuring Segment Routing MPLS IGP for MPLS Ping and Traceroute

```
ping mpls ipv4 <prefix/prefix_length> [fec-type [ldp | bgp | generic | isis | ospf]]
[sr-path-type [ip | sid | strict-sid]]
[destination <addr_start> [<addr_end> [<addr_incr_mask> | <addr_incr>]]]
[source <addr>]
[exp <exp-value>]
[pad <pattern>]
[ttl <ttl>]
[reply [mode [ipv4 | router-alert | no-reply]]
[dscp <dscp-bits>]
[pad-tlv]]
[verbose]
[output {interface <tx-interface>} [nexthop <nexthop ip addr>]]
[{dsmap | ddmap [l2ecmp]} [hashkey {none | {ipv4 | ipv4-label-set {bitmap <bitmap_size>}}]

traceroute mpls [multipath] ipv4 <prefix/prefix_length> [fec-type [ldp | bgp | generic |
isis | ospf]] [sr-path-type [ip | sid | strict-sid]]
[timeout <seconds>]
[destination <addr_start> [<addr_end> [<addr_incr_mask> | <addr_incr>]]]
[source <addr> ]
[exp <exp-value>]
[ttl <ttl-max>]
[reply [mode [ipv4 | router-alert | no-reply]]
[pad-tlv]]
[output {interface <tx-interface>} [nexthop <nexthop ip addr>]]
[flags {fec | ttl}]
[hashkey ipv4 | ipv4-label-set {bitmap <bitmap_size>}]
```

- fec-type: IPv4 Target FEC type, use head end auto detected FEC type by default.
- sr-path-type: Segment routing path type selection algorithm. Use IP imposition path, when option is specified.

Note On IOS XE platforms, **traceroute mpls ipv4** <*prefix*> uses LDP labels by default, even if **sr-label-preferred** command is configured. To use SR labels, specify the FEC type in the command. This behavior is the same on all IOS XE platforms.



CHAPTER 16

Using Seamless BFD with Segment Routing

The Segment Routing TE feature provides information support for Seamless Bidirectional Forwarding Detection (S-BFD).

- Feature Information for Seamless BFD with Segment Routing, on page 161
- Restrictions For Using Seamless BFD with Segment Routing, on page 162
- Information About Seamless BFD with Segment Routing, on page 162
- How to Configure Seamless BFD with Segment Routing, on page 163
- Additional References for Seamless BFD with Segment Routing, on page 165

Feature Information for Seamless BFD with Segment Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 14: Feature Information for Segment Routing TE Feature

Feature Name	Releases	Feature Information
Segment Routing TE Feature	Cisco IOS XE Amsterdam 17.3.2	Seamless Bidirectional Forwarding Detection (S-BFD), is a simplified mechanism for using BFD with a large proportion of negotiation aspects eliminated, thus providing benefits such as quick provisioning, as well as improved control and flexibility for network nodes initiating path monitoring. The following commands were introduced or modified: **address-family ipv4 strict-spf**, **bfd-template single-hop**, **index range**, **sbfd local-discriminator**, **show bfd neighbor**, **show isis segment-routing**, **show mpls forwarding-table**, **show mpls traffic tunnel**, **show mpls traffic-engineering**.

Restrictions For Using Seamless BFD with Segment Routing

Restrictions for Seamless-Birdirectional Forwarding (S-BFD)

- Seamless-Birdirectional Forwarding (S-BFD) supporting IPv4 only for segment routing traffic engineering (SR-TE). IPv6 is not supported.
- Single hop S-BFD session is only supported.
- RSVP-TE does not support S-BFD.

Information About Seamless BFD with Segment Routing

Bidirectional Forwarding Detection and Seamless-Bidirectional Forwarding Detection (S-BFD)

Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols.

BFD provides a consistent failure detection method for network administrators, in addition to fast forwarding path failure detection. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable.

Seamless Bidirectional Forwarding Detection (S-BFD), is a simplified mechanism for using BFD with a large proportion of negotiation aspects eliminated, thus providing benefits such as quick provisioning, as well as improved control and flexibility for network nodes initiating path monitoring.

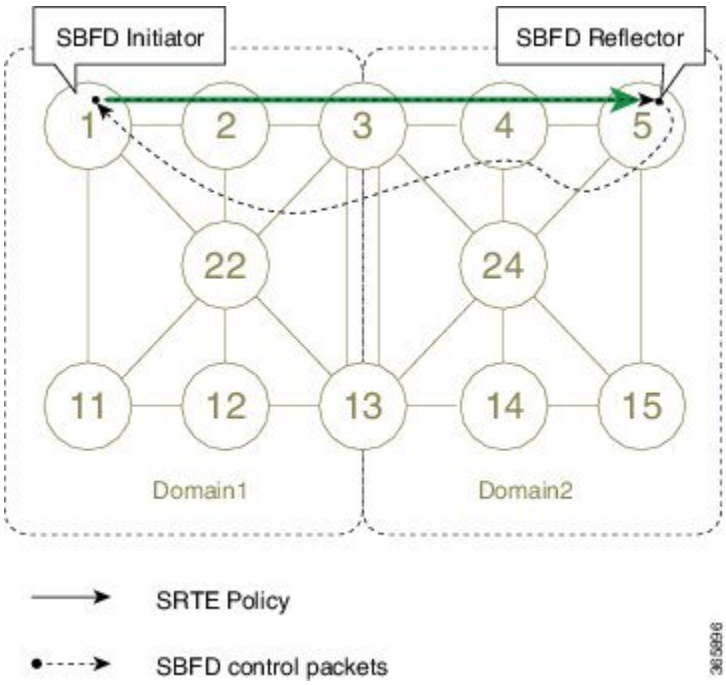
If SBFD session fails, S-BFD brings down the SR-TE session. S-BFD also provides faster session bring up due to less control packets exchange. S-BFD is associated with SR-TE to bring a session up quickly. The BFD state is only maintained at head end thereby reducing overhead.

S-BFD implements support for RFC 7880, RFC 7881 on segment routing.

Initiators and Reflectors

SBFD runs in an asymmetric behavior, using initiators and reflectors. The following figure illustrates the roles of an SBFD initiator and reflector.

Figure 19: SBFD Initiator and Reflector



The initiator is an SBFD session on a network node that performs a continuity test to a remote entity by sending SBFD packets. The initiator injects the SBFD packets into the segment-routing traffic-engineering (SRTE) policy. The initiator triggers the SBFD session and maintains the BFD state and client context.

The reflector is an SBFD session on a network node that listens for incoming SBFD control packets to local entities and generates response SBFD control packets. The reflector is stateless and only reflects the SBFD packets back to the initiator.

A node can be both an initiator and a reflector, thereby allowing you to configure different SBFD sessions.

S-BFD can be enabled and supported for SR-TE IPv4, but IPv6 is not supported. For SR-TE, S-BFD control packets are label switched in forward and reverse direction. For S-BFD, the tail end is the reflector node. Other nodes cannot be a reflector. When using S-BFD with SR-TE, if the forward and return directions are label switched paths, S-BFD need not be configured on the reflector node.

How to Configure Seamless BFD with Segment Routing

Configuring Seamless-Bidirectional Forwarding Detection (S-BFD) for Segment Routing

S-BFD must be enabled on both initiator and reflector nodes.

Note When using S-BFD with SR-TE, if the forward and return directions are label switched paths, S-BFD need not be configured on the reflector node.

Enabling Seamless Bidirectional Forwarding Detection (S-BFD) on the Reflector Node

Perform this task to configure S-BFD on the reflector node.

```
sbfd local-discriminator 10.55.55.55
```

Enabling Seamless Bidirectional Forwarding Detection (S-BFD) on the Initiator Node

Perform this task to configure S-BFD on the initiator node.

```
bfd-template single-hop ABC
 interval min-tx 300 min-rx 300 multiplier 10
```

Enabling Segment Routing Traffic Engineering Tunnel with Seamless-Bidirectional Forwarding (S-BFD)

```
interface Tunnel56
 ip unnumbered Loopback11
 tunnel mode mpls traffic-eng
 tunnel destination 10.55.55.55 */IP address of Reflector node/*
 tunnel mpls traffic-eng path-option 1 dynamic segment-routing
 tunnel mpls traffic-eng bfd sbfd ABC
!
end
```

Verifying S-BFD Configuration

SUMMARY STEPS

1. **show mpls traffic-engineering tunnel** *tunnel-name*
2. **show bfd neighbors**

DETAILED STEPS

Procedure

Step 1 **show mpls traffic-engineering tunnel** *tunnel-name*

Verifies the SR TE state and the S-BFD session state.

Example:

```
Router# sh mpls traffic-eng tunnel tunnel 56

    Name: R1_t56                             (Tunnel56) Destination: 10.55.55.55
      Status:
        Admin: up         Oper: up     Path: valid       Signalling: connected
        path option 1, (SEGMENT-ROUTING) type dynamic (Basis for Setup, path weight 12)

      Config Parameters:
        Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
        Metric Type: TE (default)
        Path Selection:
         Protection: any (default)
        Path-selection Tiebreaker:
          Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
        Hop Limit: disabled
```

```
  Cost Limit: disabled
  Path-invalidation timeout: 10000 msec (default), Action: Tear
  AutoRoute: disabled LockDown: disabled Loadshare: 0 [0] bw-based
  auto-bw: disabled
  Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No

SBFD configured with template: ABC
  Session type: CURRENT        State: UP       SBFD handle: 0x3
  LSP ID: 1
  Last uptime duration: 3 minutes, 35 seconds
  Last downtime duration: --
    Active Path Option Parameters:
  State: dynamic path option 1 is active
  BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
Node Hop Count: 2
    History:
  Tunnel:
    Time since created: 4 minutes, 3 seconds
    Number of LSP IDs (Tun_Instances) used: 1
  Current LSP: [ID: 1]
    Uptime: 3 minutes, 36 seconds
Tun_Instance: 1
Segment-Routing Path Info (isis  level-2)
  Segment0[Link]: 10.12.12.1 - 10.12.12.2, Label: 48
  Segment1[Link]: 10.25.25.2 - 10.25.25.5, Label: 35 !
```

Step 2 **show bfd neighbors**

Verifies that BFD neighbors are established properly.

Example:

```
Router# show bfd neighbors

    MPLS-TE SR Sessions
    Interface      LSP ID(Type)                  LD/RD         RH/RS     State
    Tunnel56          1 (SR)                  4097/926365495  Up        Up
```

Additional References for Seamless BFD with Segment Routing

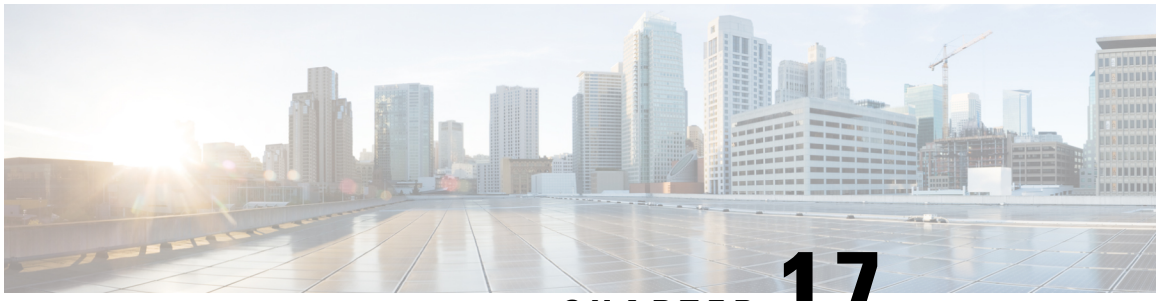
Related Documents

Related Topic	Document Title
Segment Routing Traffic Engineering configuration	*Segment Routing -Traffic Engineering*

Table 15: Standards and RFC

Standard/RFC	Title
draft-akiya-bfd-seamless-base-03	Seamless Bidirectional Forwarding Detection (S-BFD)
draft-ietf-isis-segment-routing-extensions-07	IS-IS Extensions for Segment Routing
draft-ietf-spring-segment-routing-09	Segment Routing Architecture

Standard/RFC	Title
RFC 7880	Seamless Bidirectional Forwarding Detection (S-BFD)
RFC 7881	Seamless Bidirectional Forwarding Detection (S-BFD) for IPv4, IPv6, and MPLS



CHAPTER 17

Using SSPF with Segment Routing

The Segment Routing TE feature provides information support for the Strict Shortest Path First (SPF).

- Feature Information for SSPF with Segment Routing, on page 167
- Information About SSPF with Segment Routing, on page 167
- How to Configure SSPF with Segment Routing, on page 168
- Additional References for SSPF with Segment Routing, on page 170

Feature Information for SSPF with Segment Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 16: Feature Information for Segment Routing SSPF Feature

Feature Name	Releases	Feature Information
Segment Routing TE Feature	Cisco IOS XE Amsterdam 17.3.2	The Segment Routing TE feature provides information support for the Strict Shortest Path First (SPF).. The following commands were introduced or modified: **address-family ipv4 strict-spf**, **bfd-template single-hop**, **index range**, **sbfd local-discriminator**, **show bfd neighbor**, **show isis segment-routing**, **show mpls forwarding-table**, **show mpls traffic tunnel**, **show mpls traffic-engineering**.

Information About SSPF with Segment Routing

Strict Shortest Path First

Segment Routing supports the following two algorithms:

- Algorithm 0: This is a Shortest Path First (SPF) algorithm based on link metric. This shortest path algorithm is computed by the Interior gateway protocol (IGP).
- Algorithm 1: This is a Strict Shortest Path First (SSPF) algorithm based on link metric. The algorithm 1 is identical to algorithm 0 but requires that all nodes along the path honor the SPF routing decision. Local policy does not alter the forwarding decision. For example, a packet is not forwarded through locally engineered path.

Different SIDs are associated with the same prefix for each algorithm.

Strict Shortest Path First is supported by default - but strict SIDs must be configured for at least one node address on each node supporting Segment Routing.

Approaches for Configure Strict Shortest Path First

The two approaches to configure Strict SFP are as follows:

- Using the **connect-prefix-sid-map** command—Strict SFP is configured globally on all the nodes. For a network to be Strict SFP-aware (that is, for ISIS to populate Strict SPF), all nodes must be configured with a local Strict SFP SID.
- Using Segment-routing Mapping Server—One node in the network is configured as mapping server and the remaining nodes act as a client.

How to Configure SSPF with Segment Routing

Configuring Strict Shortest Path First (SPF)

Enabling Strict Shortest Path First Using the connect-prefix-sid-map command

Enabling Shortest Path First on a Provider-Edge Device

When enabling Strict Shortest Path First using the **connect-prefix-sid-map** command, the Strict Shortest Path First (SPF) must be configured on the provider-edge device first and then on the node devices. The following is a sample configuration code snippet to enable Strict Shortest Path First on a provider-edge device.

```
segment-routing mpls
connected-prefix-sid-map
  address-family ipv4
   10.10.10.10/32 index 100 range 1
  exit-address-family
  address-family ipv4 strict-spf
   10.10.10.10/32 index 1000 range 1 ------------------configure strict SPF locally
  exit-address-family
```

Enabling Shortest Path First on a Node Device

The following is a sample configuration code snippet to enable Strict Shortest Path First on a node in the network and must be enabled on all nodes in a network.

```
segment-routing mpls
```

```
connected-prefix-sid-map
  address-family ipv4
   10.20.20.20/32 index 110 range 1
  exit-address-family
  address-family ipv4 strict-spf
   10.20.20.20/32 index 1100 range 1
  exit-address-family
```

Enabling Strict Shortest Path First Using Segment Routing Mapping Server

Configuring a Node as Segment Routing Mapping Server

The following is a sample configuration code snippet to configure a node as Segment Routing Mapping Server.

```
segment-routing mpls
mapping-server
  prefix-sid-map
   address-family ipv4
    10.10.10.10/32 index 100 range 1
    10.20.20.20/32 index 110 range 1
    10.30.30.30/32 index 120 range 1
    10.40.40.40/32 index 130 range 1
    10.50.50.50/32 index 140 range 1
   exit-address-family
   address-family ipv4 strict-spf
    10.10.10.10/32 index 1000 range 1
    10.20.20.20/32 index 1100 range 1
    10.30.30.30/32 index 1200 range 1
    10.40.40.40/32 index 1300 range 1
    10.50.50.50/32 index 1400 range 1
    10.100.100.100/32 index 2000 range 1
   exit-address-family
```

Configuring the Segment Routing Mapping Server to Advertise and Receive Local Prefixes

The following is a sample configuration code snippet to configure a Segment Routing Mapping Server to advertise and receive local prefixes.

```
router isis SR
segment-routing mpls
 segment-routing prefix-sid-map advertise-local
 segment-routing prefix-sid-map receive
```

Verifying ISIS Advertises the SIDs

The following is a sample configuration code snippet to verify that ISIS advertises the SIDs.

```
Router# show isis segment-routing prefix-sid-map advertise strict-spf
Tag SR:
IS-IS Level-1 advertise prefix-sid maps:
Prefix               SID Index    Range        Flags
10.10.10.10/32       1000         1
10.20.20.20/32       1100         1
10.30.30.30/32       1200         1
10.40.40.40/32       1300         1
10.50.50.50/32       1400         1
10.100.100.100/32    2000         1
Tag SR:
IS-IS Level-2 advertise prefix-sid maps:
Prefix               SID Index    Range        Flags
10.10.10.10/32       1000         1
10.20.20.20/32       1100         1
10.30.30.30/32       1200         1
```

```
10.40.40.40/32       1300         1
10.50.50.50/32       1400         1
10.100.100.100/32    2000         1
```

The following is a sample configuration code snippet to verify that a provider-edge device receives Strict Shortest Path First SID from SRMS Server.

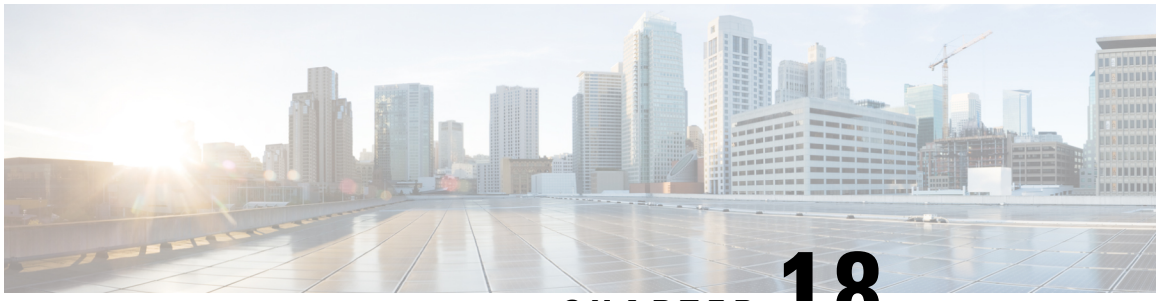
```
Router# show isis segment-routing prefix-sid-map receive strict-spf

Tag SR:
IS-IS Level-1 receive prefix-sid maps:
Host                  Prefix              SID Index    Range        Flags
P1                    10.10.10.10/32      1000         1
                      10.20.20.20/32      1100         1
                      10.30.30.30/32      1200         1
                      10.40.40.40/32      1300         1
                      10.50.50.50/32      1400         1
                      10.100.100.100/32   2000         1
Tag SR:
IS-IS Level-2 receive prefix-sid maps:
Host                  Prefix              SID Index    Range        Flags
P1                    10.10.10.10/32      1000         1
                      10.20.20.20/32      1100         1
                      10.30.30.30/32      1200         1
                      10.40.40.40/32      1300         1
                      10.50.50.50/32      1400         1
                      10.100.100.100/32   2000         1
```

Additional References for SSPF with Segment Routing

Related Documents

Related Topic	Document Title
Segment Routing Traffic Engineering configuration	*Segment Routing -Traffic Engineering*



CHAPTER 18

Dynamic PCC

The Stateful Path Computation Element Protocol(PCEP) enables a router to report and optionally delegate Label Switched Paths (LSPs) which is established using either Resource Reservation Protocol (RSVP) protocol or Segment Routing Traffic Engineering (SR-TE) to a stateful Path Computation Element (PCE).

An LSP delegated to a PCE can be updated by the PCE and a stateful PCE can compute and provide the path of an LSP to the Path Computation Client (PCC).

SR-TE and RSVP-TE LSPs require link-state routing protocols such as OSPF or ISIS to distribute and learn traffic engineering topology. A stateful PCE can learn the traffic engineering topology through BGP Link-State protocol. You can use the verbatim path option in the case when all or some of the intermediate nodes in a network do not support IGP extensions for TE.

- Information About Dynamic PCC, on page 171
- How to Configure Dynamic PCC, on page 172
- Verifying Dynamic PCC, on page 173
- Verifying Verbatim Path Option With Dynamic PCC, on page 176
- Feature Information for Dynamic PCC, on page 177

Information About Dynamic PCC

Path Computation Element Protocol Functions

A Path Computation Element Protocol (PCEP) session is a TCP session between a PCC and a PCE with protocol messages. The PCEP functions are verified based on the PCC functions. The configuration and verification show that the request is accepted and path computation is provided based on PCReq message from the client. The passive reporting enables a router to report a tunnel instead of delegating it to a PCE. The PCE is aware of the tunnel even though it cannot modify the tunnel.

PCEP functions are useful when a network has both router-controlled and PCE delegated tunnels. The PCE is aware of both the tunnels and can make an accurate decision on path computation.

Redundant Path Computation Elements

For redundancy it may be required to deploy redundant PCE servers. A PCC uses precedence to select stateful PCEs for delegating LSPs. Precedence can take any value between 0 and 255. The default precedence value is 255. When there are multiple stateful PCEs with active PCEP session, PCC chooses the PCE with the lowest

precedence value. In case where primary PCE server session goes down, PCC router re-delegates all tunnels to next available PCE server. You can use the following CLIs in the case of redundant PCEs:

```
R2(config)#mpls traffic-eng pcc peer 10.77.77.77 source 10.22.22.22 precedence 255
R2(config)#mpls traffic-eng pcc peer 10.88.88.88  source 10.22.22.22 precedence 100
!
end
```

In the above example PCE server with IP address 10.88.88.88 is the primary PCE server since it has lower precedence value.

How to Configure Dynamic PCC

Configuring Dynamic PCC Globally

Perform the following task to configure dynamic PCC globally

```
enable
configure terminal
mpls traffic-eng tunnels
mpls traffic-eng pcc peer 10.0.0.1    ----□(10.0.0.1 is the PCE server address)
mpls traffic-eng pcc report-all
end
```



Note **mpls traffic-eng pcc report-all** is not mandatory for PCE/PCC basic operational delegated tunnels. It is required to report locally calculated LSPs to the PCE server.

Configuring Dynamic PCC on an Interface

Perform the following task to configure dynamic PCC on an interface

```
interface Tunnel1
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 10.7.7.7
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 5 5
 tunnel mpls traffic-eng bandwidth 200
 tunnel mpls traffic-eng path-option 10 dynamic pce segment-routing
end
```

Configuring Dynamic PCC With Verbatim Path Option

To enable Dynamic PCC with verbatim path option, use the following CLI under the SR-TE tunnel interface:

```
R1#
interface Tunnel2
ip unnumbered Loopback11
```

```
tunnel mode mpls traffic-eng
tunnel destination 10.66.66.66
tunnel mpls traffic-eng autoroute destination
tunnel mpls traffic-eng path-option 1 dynamic segment-routing pce verbatim
```

Verifying Dynamic PCC

The following sample output is from the **show pce client peer detail** command.

```
Device# show pce client peer detail

PCC's peer database:
--------------------

Peer address: 10.1.1.1
  State up
  Capabilities: Stateful, Update, Segment-Routing
  PCEP has been up for: 23:44:58
  PCEP session ID: local 1, remote: 0
  Sending KA every 30 seconds
  Minimum acceptable KA interval: 20 seconds
  Peer timeout after 120 seconds
  Statistics:
    Keepalive messages: rx    2798 tx    2112
    Request messages:   rx       0 tx      32
    Reply messages:     rx      32 tx       0
    Error messages:     rx       0 tx       0
    Open messages:      rx       1 tx       1
    Report messages:    rx       0 tx      57
    Update messages:    rx      72 tx       0
```

The following sample output is from the **show mpls traffic-eng tunnels tunnel 1** command which shows the LSP details.

```
Device# show mpls traffic-eng tunnels tunnel 1

Name: d1_t1                           (Tunnel1) Destination: 10.7.7.7
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 10, (SEGMENT-ROUTING) (PCE) type dynamic (Basis for Setup, path weight 0)

  Config Parameters:
    Bandwidth: 200      kbps (Global)  Priority: 5  5   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    Path Selection:
     Protection: any (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
    Hop Limit: disabled
    Cost Limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear
    AutoRoute: enabled  LockDown: disabled Loadshare: 200 [10000000] bw-based
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: dynamic path option 10 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled

  PCEP Info:
```

```
    Delegation state: Working: yes   Protect: no
    Current Path Info:
      Request status: processed
      Created via PCRep message from PCE server: 10.1.1.1
    Reported paths:
      Tunnel Name: csr551_t2001
       LSPs:
        LSP[0]:
         source 10.2.2.2, destination 10.7.7.7, tunnel ID 1, LSP ID 5
         State: Admin up, Operation active
         Setup type: SR
         Bandwidth: signaled 0
         LSP object:
           PLSP-ID 0x807D1, flags: D:0 S:0 R:0 A:1 O:2
         Reported path:
           Metric type: TE, Accumulated Metric 0

  History:
    Tunnel:
      Time since created: 34 minutes, 3 seconds
      Time since path change: 1 minutes, 44 seconds
      Number of LSP IDs (Tun_Instances) used: 5
    Current LSP: [ID: 5]
      Uptime: 1 minutes, 44 seconds
    Prior LSP: [ID: 3]
      ID: path option unknown
      Removal Trigger: path verification failed
  Tun_Instance: 5
  Segment-Routing Path Info (isis  level-1)
    Segment0[Node]: 10.3.3.3, Label: 20270
    Segment1[Node]: 10.6.6.6, Label: 20120
    Segment2[Node]: 10.7.7.7, Label: 20210
```

The following sample output is from the **show pce client lsp detail** command.

```
Device# show pce client lsp detail

PCC's tunnel database:
----------------------
Tunnel Name: d1_t1
 LSPs:
  LSP[0]:
   source 10.2.2.2, destination 10.7.7.7, tunnel ID 1, LSP ID 5
   State: Admin up, Operation active
   Setup type: SR
   Bandwidth: signaled 0
   LSP object:
     PLSP-ID 0x807D1, flags: D:0 S:0 R:0 A:1 O:2
   Reported path:
     Metric type: TE, Accumulated Metric 0
```

The following sample output is from the **show pce lsp detail** command which shows the tunnel is delegated.

```
Device# show pce lsp detail

Thu Jul  7 10:24:30.836 EDT

PCE's tunnel database:
----------------------
PCC 10.103.2.1:
```

```
Tunnel Name: d1_t1
 LSPs:
  LSP[0]:
   source 10.2.2.2, destination 10.7.7.7, tunnel ID 1, LSP ID 5
   State: Admin up, Operation active
   Binding SID: 0
   PCEP information:
     plsp-id 526289, flags: D:1 S:0 R:0 A:1 O:2
   Reported path:
     Metric type: TE, Accumulated Metric 0
      SID[0]: Node, Label 20270, Address 10.3.3.3
      SID[1]: Node, Label 20120, Address 10.6.6.6
      SID[2]: Node, Label 20210, Address 10.7.7.7
   Computed path:
     Metric type: TE, Accumulated Metric 30
      SID[0]: Node, Label 20270, Address 10.3.3.3
      SID[1]: Node, Label 20120, Address 10.6.6.6
      SID[2]: Node, Label 20210, Address 10.7.7.7
   Recorded path:
     None
```

The following sample output is from the **show pce client lsp detail** command for reported tunnel.

```
Device# show pce client lsp detail

PCC's tunnel database:
----------------------
Tunnel Name: d1_t2
 LSPs:
  LSP[0]:
   source 10.2.2.2, destination 10.7.7.7, tunnel ID 2, LSP ID 1
   State: Admin up, Operation active
   Setup type: SR
   Bandwidth: signaled 0
   LSP object:
     PLSP-ID 0x807D2, flags: D:0 S:0 R:0 A:1 O:2
   Reported path:
     Metric type: TE, Accumulated Metric 30
```

The following sample output is from the **show pce lsp detail** command which shows the tunnel is not delegated.

```
Device# show pce lsp detail

Thu Jul  7 10:29:48.754 EDT

PCE's tunnel database:
----------------------
PCC 10.0.0.1:

Tunnel Name: d1_t2
 LSPs:
  LSP[0]:
   source 10.2.2.2, destination 10.7.7.7, tunnel ID 2, LSP ID 1
   State: Admin up, Operation active
   Binding SID: 0
   PCEP information:
     plsp-id 526290, flags: D:0 S:0 R:0 A:1 O:2
   Reported path:
     Metric type: TE, Accumulated Metric 30
      SID[0]: Adj, Label 74, Address: local 172.16.0.1 remote 172.16.0.2
      SID[1]: Adj, Label 63, Address: local 172.17.0.1 remote 172.17.0.2
```

```
      SID[2]: Adj, Label 67, Address: local 172.18.0.1 remote 172.18.0.2
      SID[3]: Node, Label unknownAddress 10.7.7.7
   Computed path:
     None
   Recorded path:
     None
```

Verifying Verbatim Path Option With Dynamic PCC

To verify proper operation with verbatim path option, use the following command:

```
R1#sh mpl tr tun tun 2
Name: R1_t2                                (Tunnel2) Destination: 10.66.66.66
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, (SEGMENT-ROUTING) (PCE) type dynamic (verbatim) (Basis for Setup)

  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (interface)
    Path Selection:
     Protection: any (default)
    Path-selection Tiebreaker:
      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)
    Hop Limit: disabled [ignore: Verbatim Path Option]
    Cost Limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear
    AutoRoute: disabled LockDown: disabled Loadshare: 0 [0] bw-based
    AutoRoute destination: enabled
    auto-bw: disabled
    Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
  Active Path Option Parameters:
    State: dynamic path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: enabled

  PCEP Info:
    Delegation state: Working: yes   Protect: no
    Delegation peer: 10.77.77.77
    Working Path Info:
      Request status: processed
      Created via PCRep message from PCE server: 10.77.77.77
      PCE metric: 4, type: TE
    Reported paths:
      Tunnel Name: Tunnel2_w
       LSPs:
        LSP[0]:
         source 10.11.11.11, destination 10.66.66.66, tunnel ID 2, LSP ID 1
         State: Admin up, Operation active
         Binding SID: 17
         Setup type: SR
         Bandwidth: requested 0, used 0
         LSP object:
           PLSP-ID 0x80002, flags: D:0 S:0 R:0 A:1 O:2
         ERO:
           SID[0]: Adj, Label 24, NAI: local 10.12.12.1 remote 10.12.12.2
           SID[1]: Adj, Label 26, NAI: local 10.25.25.2 remote 10.25.25.5
           SID[2]: Adj, Label 22, NAI: local 10.56.56.5 remote 10.56.56.6

  History:
    Tunnel:
```

```
      Time since created: 39 days, 19 hours, 9 minutes
      Time since path change: 1 minutes, 3 seconds
      Number of LSP IDs (Tun_Instances) used: 1
    Current LSP: [ID: 1]
      Uptime: 1 minutes, 3 seconds
  Tun_Instance: 1
  Segment-Routing Path Info (IGP information is not used)
    Segment0[Link]: 10.12.12.1 - 10.12.12.2, Label: 24
    Segment1[Link]: 10.25.25.2 - 10.25.25.5, Label: 26
    Segment2[Link]: 10.56.56.5 - 10.56.56.6, Label: 22
!
end
```

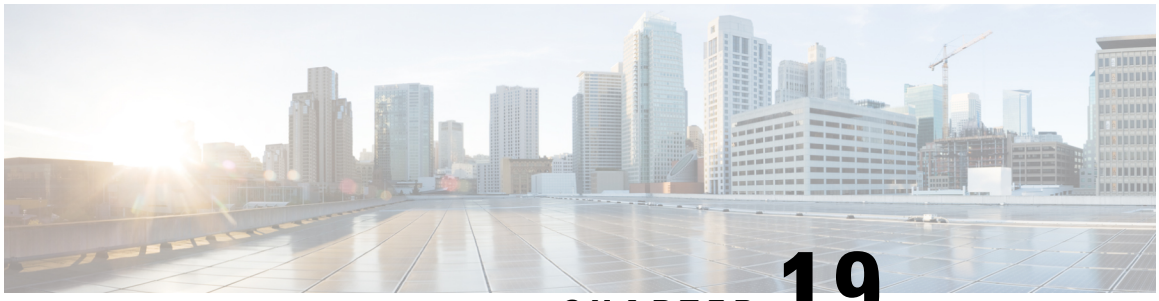
Feature Information for Dynamic PCC

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 17: Feature Information for Dynamic PCC

Feature Name	Releases	Feature Information
Dynamic PCC	Cisco IOS XE Amsterdam 17.3.2	The Dynamic Path Computation Client (PCC) feature supports an LSP delegated to a Path Computation Element (PCE).Dynamic PCC aupports both RSVP-TE and SR-TE. The following commands were added or modified: **show pce client peer detail**, **show mpls traffic-eng tunnels tunnel 1**, **show pce client lsp detail**, **show pce lsp detail**.



CHAPTER 19

SR: PCE Initiated LSPs

The SR: PCE Initiated LSPs feature provides support for PCE-initiated LSPs in stateful PCE model on segment routing networks.

- Prerequisites for SR: PCE Initiated LSPs, on page 179
- Restrictions for SR: PCE Initiated LSPs, on page 179
- Information About SR: PCE Initiated LSPs, on page 179
- How to Configure SR: PCE Initiated LSPs, on page 181
- Additional References for SR: PCE Initiated LSPs, on page 187
- Feature Information for SR: PCE Initiated LSPs, on page 187

Prerequisites for SR: PCE Initiated LSPs

- The Dynamic PCC feature must be configured.
- Auto tunnels must be enabled on the PCC.

Restrictions for SR: PCE Initiated LSPs

- The SR: PCE Initiated LSPs feature supports only basic LSP generation and does not support TE attributes.

Information About SR: PCE Initiated LSPs

Overview of Path Computation Element Protocol

draft-ietf-pce-stateful-pce-21 describes Stateful Path Computation Element Protocol (PCEP) enables a router to report and optionally delegate Label Switched Paths (LSPs) which is established using either Resource Reservation Protocol (RSVP) protocol or Segment Routing Traffic Engineering (SR-TE) to a stateful Path Computation Element (PCE). An LSP delegated to a PCE can be updated by the PCE and a stateful PCE can compute and provide the path of an LSP to the Path Computation Client (PCC).

The **PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model (draft-ietf-pce-pce-initiated-lsp-11)** specifies a set of extensions to PCEP to enable stateful control of TE

LSPs across PCEP sessions in compliance with RFC4657. The **PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model** provides information about the following:

- Configuring LSPs on a PCC
- Delegating control of LSP to a PCE

SR: PCE Initiated LSPs

The SR: PCE Initiated LSPs feature allows a client to create, setup, control, and delete an LSP from PCE server, which controls creating and deleting LSP on PCC through an PCE initiate message. PCE initiated LSP is automatically delegated to the PCE server that initiated the LSP. A PCE client processes an LSP initiate message. By using the LSP initiate message, PCE client can create or delete LSP.

When a failover occurs on a route processor (RP), the failover results in the RP being disconnected from the network. To reestablish the connection, the PCE server has to resend LSP initiate message to reclaim PCE Initiated LSPs on a client, else PCE initiated LSPs created by the client are automatically deleted.

You must use the **pce** command for establishing a PCEP session with PCC. The **force auto-route** command is used to advertise an LSP within an area via the autoroute announce message and across areas via the autoroute destination message. The decision to use autoroute announce or autoroute destination is performed by a device depending on the destination IP address. Enabling the **force auto-route** command for an initiated LSP allows automatic routing of traffic through a TE tunnel instead of routing traffic via manually configuring static routes. The autoroute announce message installs routes announced by the destination router and downstream routers into the routing table of a headend device that can be reached through a tunnel.

The PCC configuration includes IP addresses for each PCE (both primary and standby or more). The precedence for each PCE can be explicitly specified. If the precedence for two PCEs is same, PCE with smaller IP address has a higher precedence.

Single and Redundant PCE Operations

The SR: PCE Initiated LSPs feature supports single and redundant PCE operations. In a single PCE operation, when a PCE fails, PCC waits until the state timeout expiry (60 seconds) to remove the LSP.

In a redundant PCE operation, if a Representational state transfer (REST) call is initiated to a standby PCE before the expiry of the timer, the initiated LSP is retained else, the LSP is removed.



Note The REST call must be initiated again to a standby PCE if the primary PCE fails, and the call must include the standby PCE IP address.

In a redundant PCE operation, PCC configurations include both primary and standby IP addresses for an LSP and the IP address with a lower precedence becomes the primary PCE. The IP addreseses are compared incase of equal priority.

How to Configure SR: PCE Initiated LSPs

Establishing a PCEP session with PCC

Perform this task to configure a PCEP session PCE server XR based XTC server.

```
configure terminal
pce
 address ipv4 192.0.2.1
end
```

The IP address 192.0.2.1 is the IP address of the transport controller.

Advertising an LSP in a Network

```
configure terminal
 mpls traffic-eng pcc peer 192.0.2.1 source 203.0.113.1 force-autoroute
 end
```

In the above code snippet, 192.0.2.1 is PCE IP address and 203.0.113.1 is PCC source address for establishing a PCEP session.

Specifying Precedence of a PCE for PCC

```
configure terminal
mpls traffic-eng pcc peer 192.0.2.1 source 203.0.113.1 force autoroute precedence 255
mpls traffic-eng pcc peer 192.0.2.2 source 203.0.113.1 force-autoroute precedence 100
end
```

In the above code snippet, 100 is a lower precedence than 255, which is the default precedence. Therefore, the device with IP address 192.0.2.2 becomes the primary PCE and the device with 192.0.2.1 becomes the standby PCE.

Triggering PCE server precedence re-evaluation

A change in a PCE server's precedence is not considered a PCE server failure. So, the change in precedence does not trigger a redelegation timeout or a re-evaluation of LSP delegation to the PCE server at a PCC.

Re-evaluation of LSP delegation to PCE servers after CLI reconfiguration is controlled by the TE reoptimisation timer. By default, the TE reoptimisation timer is set to 3600 seconds.

You can accelerate the re-evaluation of LSP delegation from a PCC to PCE servers after you have changed the precedence of PCE servers or added new PCE servers. To do so, manually trigger TE reoptimisation using the following command in privileged EXEC mode:

```
mpls traffic-eng reoptimize
```

Verifying LSP Configurations

SUMMARY STEPS

1. **show pce ipv4 peer detail**
2. **show pce lsp detail**
3. **show pce client peer**
4. **show mpls traffic-eng tunnel tunnel** *number*

DETAILED STEPS

Procedure

Step 1 **show pce ipv4 peer detail**

Use this command to verify PCEP session details on a PCE. In this example, the term instantiation indicates that PCE supports initiated LSP.

```
Device# show pce ipv4 peer detail


PCE's peer database:

--------------------

Peer address: 10.2.2.2----' PCC IP address

                State: Up

  Capabilities: Stateful, Segment-Routing, Update, Instantiation
```

Step 2 **show pce lsp detail**

Use this command to verify the initiated LSP on a PCE.

```
Device# show pce lsp detail


PCE's tunnel database:

----------------------

PCC 10.52.2.2 ----' PCC IP address


Tunnel Name: Test1--------' tunnel name set by REST Call
```

```
LSPs:
  LSP[0]:
   source 10.52.2.2, destination 10.57.7.7, tunnel ID 2000, LSP ID 1
   State: Admin up, Operation active
   Binding SID: 26
   PCEP information:
     plsp-id 526288, flags: D:1 S:0 R:0 A:1 O:2 C:1
   LSP Role: Single LSP
   State-sync PCE: None
   PCC: 10.52.2.2
   LSP is subdelegated to: None
   Reported path:
     Metric type: TE, Accumulated Metric 2
      SID[0]: Adj, Label 25, Address: local 10.105.3.1 remote 10.105.3.2
      SID[1]: Adj, Label 24, Address: local 10.104.8.2 remote 10.104.8.1
      SID[2]: Adj, Label 38, Address: local 10.107.10.1 remote 10.107.10.2
   Computed path: (Local PCE)
     None
     Computed Time: Not computed yet
   Recorded path:
     None
   Disjoint Group Information:
     None
```

Step 3 **show pce client peer**

Use this command to verify a PCEP session output on a PCC and to verify if the **force-autoroute** command is enabled.

```
Device# show pce client peer


PCC's peer database:

------------------------------

Peer address: 10.51.1.1, Precedence: 255

  State up

  Capabilities: Stateful, Update, Segment-Routing,Force-autoroute
```

Step 4 **show mpls traffic-eng tunnel tunnel** *number*

Use this command to verify the output of the initiated LSP tunnel on a PCC.

```
Device# show mpls traffic-eng tunnel tunnel 2000


Name: Test1                                (Tunnel2000) Destination: 10.57.7.7 Ifhandle: 0x11E
(auto-tunnel for pce client)

  Status:

    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 1, (SEGMENT-ROUTING) (PCE) type dynamic (Basis for Setup)


  Config Parameters:

    Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF

    Metric Type: TE (default)

    Path Selection:

     Protection: any (default)

    Path-selection Tiebreaker:

      Global: not set   Tunnel Specific: not set   Effective: min-fill (default)

    Hop Limit: disabled
```

```
  Cost Limit: disabled
  Path-invalidation timeout: 10000 msec (default), Action: Tear
  AutoRoute: enabled  LockDown: disabled Loadshare: 0 [0] bw-based
  auto-bw: disabled
  Fault-OAM: disabled, Wrap-Protection: disabled, Wrap-Capable: No
Active Path Option Parameters:
  State: dynamic path option 1 is active
  BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled


PCEP Info:
  Delegation state: Working: yes   Protect: no
  Delegation peer: 10.51.1.1
  Working Path Info:
    Request status: delegated
    SRP-ID: 1
    Created via PCInitiate message from PCE server: 10.51.1.1-----' IP address
    PCE metric: 2, type: TE
  Reported paths:
    Tunnel Name: Test1
     LSPs:
      LSP[0]:
       source 10.52.2.2, destination 10.57.7.7, tunnel ID 2000, LSP ID 1
       State: Admin up, Operation active
      Binding SID: 26
       Setup type: SR
```

```
          Bandwidth: requested 0, used 0

          LSP object:

            PLSP-ID 0x807D0, flags: D:0 S:0 R:0 A:1 O:2

          Metric type: TE, Accumulated Metric 2

          ERO:

            SID[0]: Adj, Label 25, NAI: local 10.105.3.1 remote 10.105.3.2

            SID[1]: Adj, Label 24, NAI: local 10.104.8.2 remote 10.104.8.1

            SID[2]: Adj, Label 38, NAI: local 10.107.10.1 remote 10.107.10.2

        PLSP Event History (most recent first):

          Mon Jul 17 08:55:04.448: PCRpt update LSP-ID:1, SRP-ID:1, PST:1, METRIC_TYPE:2, REQ_BW:0,
USED_BW:0

          Mon Jul 17 08:55:04.436: PCRpt create LSP-ID:1, SRP-ID:1, PST:1, METRIC_TYPE:2, REQ_BW:0,
USED_BW:0

  History:

    Tunnel:

      Time since created: 2 hours, 42 minutes

      Time since path change: 2 hours, 42 minutes

      Number of LSP IDs (Tun_Instances) used: 1

    Current LSP: [ID: 1]

      Uptime: 2 hours, 42 minutes

  Tun_Instance: 1

  Segment-Routing Path Info (isis  level-2)

    Segment0[Link]: 10.105.3.1 - 10.105.3.2, Label: 25

    Segment1[Link]: 10.104.8.2 - 10.104.8.1, Label: 24

    Segment2[Link]: 10.107.10.1 - 10.107.10.2, Label: 38
```

Additional References for SR: PCE Initiated LSPs

Standards and RFCs

Standard/RFC	Title
draft-ietf-pce-pce-initiated-lsp-11	*PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*
RFC 5440	*Path Computation Element (PCE) Communication Protocol (PCEP)*
RFC 8231	*Path Computation Element (PCE) Communication Protocol Generic Requirements*

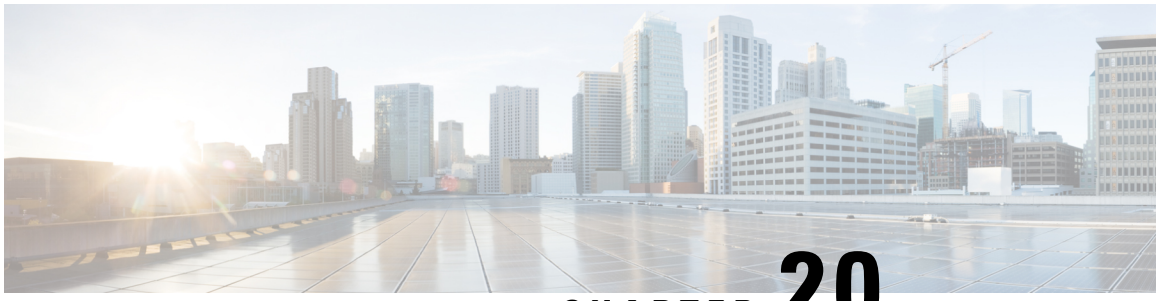
Feature Information for SR: PCE Initiated LSPs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 18: Feature Information for SR: PCE Initiated LSPs

Feature Name	Releases	Feature Information
SR: PCE Initiated LSPs	Cisco IOS XE Amsterdam 17.3.2	The SR: PCE Initiated LSPs provides support for PCE-initiated LSPs in stateful PCE model on segment routing networks. The following commands were introduced or modified: **mpls traffic-eng pcc**, **pce**, **show mpls traffic-eng tunnel**, **show pce client peer**, **show pce ipv4 peer**, **show pce lsp**.



CHAPTER 20

ISIS - SR: uLoop Avoidance

The ISIS - SR: uLoop Avoidance feature extends the ISIS Local Microloop Protection feature thereby preventing the occurrences of microloops during network convergence after a link-down event or link-up event.

- Prerequisites for ISIS - SR: uLoop Avoidance, on page 189
- Restrictions for ISIS - SR: uLoop Avoidance, on page 189
- Information About ISIS - SR: uLoop Avoidance, on page 189
- How to Enable ISIS - SR: uLoop Avoidance, on page 193
- Additional References for ISIS - SR: uLoop Avoidance, on page 194
- Feature Information for ISIS - SR: uLoop Avoidance, on page 194

Prerequisites for ISIS - SR: uLoop Avoidance

- The ISIS - SR: uLoop Avoidance feature is disabled by default. When the Topology-Independent Loop-Free Alternate (TI-LFA) feature is configured, this feature is enabled automatically. See the "Topology-Independent LFA" section in the *Using Segment Routing with IS-IS* module for more information.

Restrictions for ISIS - SR: uLoop Avoidance

- The ISIS - SR: uLoop Avoidance feature supports 2-node on the same subnet on a LAN network.

Information About ISIS - SR: uLoop Avoidance

Microloops

When changes occur in a network topology because of the failure or restoration of a link or a network device, IP Fast Reroute enables rapid network convergence by moving traffic to precomputed backup paths until regular convergence mechanisms move traffic to a newly computed best path, also known as a post-convergence path. This network convergence may cause short microloops between two directly or indirectly connected devices in the topology. Microloops are caused when different nodes in the network calculate alternate paths

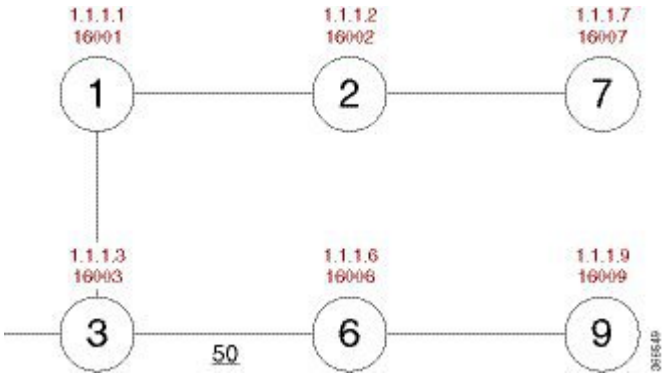
at different times and independently of each other. For instance, if a node converges and sends traffic to a neighbor node, which has not converged yet, traffic may loop between the two nodes.

Microloops may or may not result in traffic loss. If the duration of a microloop is short, that is the network converges quickly, packets may loop for a short duration before their time-to-live (TTL) expires. Eventually, the packets will get forwarded to the destination. If the duration of the microloop is long, that is one of the routers in the network is slow to converge, packets may expire their TTL or the packet rate may exceed the bandwidth, or the packets might be out of order, and packets may get dropped.

Microloops that are formed between a failed device and its neighbors are called local uloops, whereas microloops that are formed between devices that are multiple hops away are called remote uloops. Local uloops are usually seen in networks where local loop-free alternate (LFA) path is not available. In such networks, remote LFAs provide backup paths for the network.

The information discussed above can be illustrated with the help of an example topology as shown in the following figure.

Figure 20: Microloop Example Topology



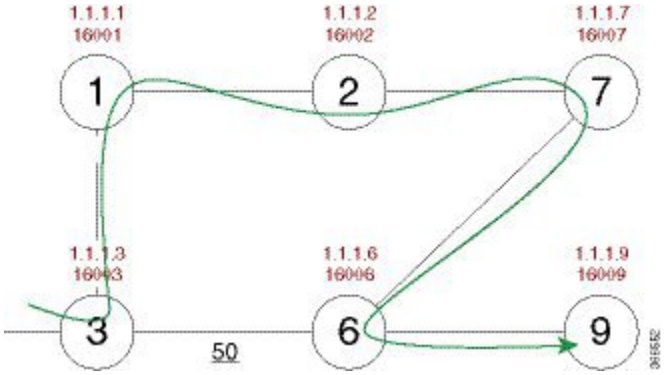
The assumptions in this example are as follows:

- The default metrics is 10 for each link except for the link between Node 3 and Node 6, which has a metric of 50. The order of convergence with SPF backoff delays on each node is as follows:
  - Node 3—50 milliseconds
  - Node 1—500 milliseconds
  - Node 2—1 second
  - Node 2—1.5 seconds

A packet sent from Node 3 to Node 9, the destination, traverses via Node 6.

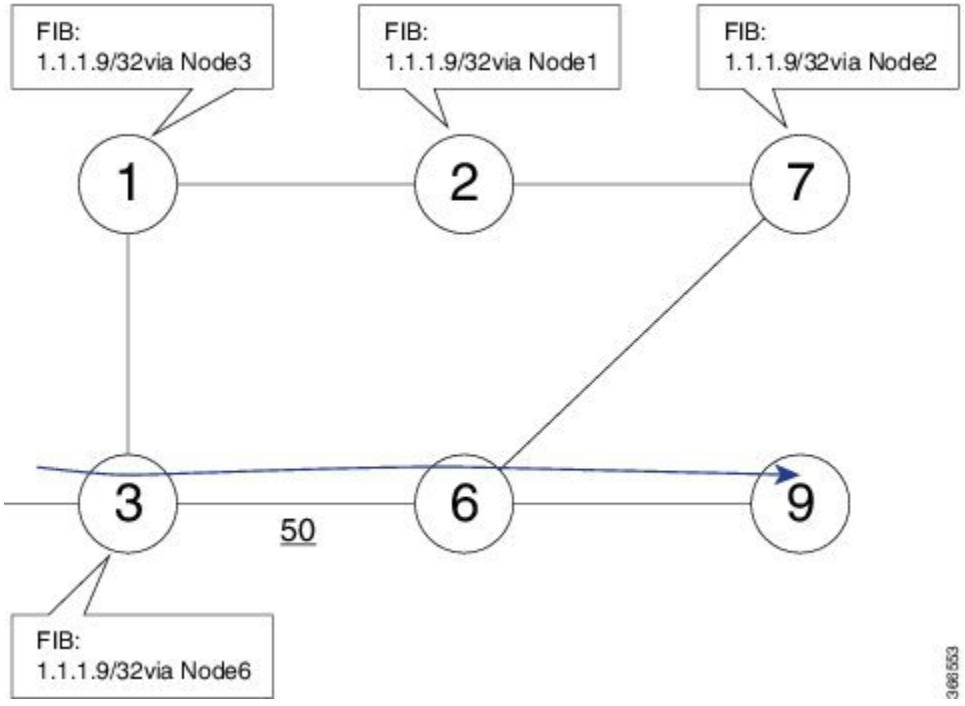
If a link is established between Node 6 and Node 7, the shortest path for a packet from Node 3 to Node 9 would be Node 1, Node 2, Node 7, and Node 6 before the packet reaches the destination, Node 9.

Figure 21: Microloop Example Topology—Shortest Path



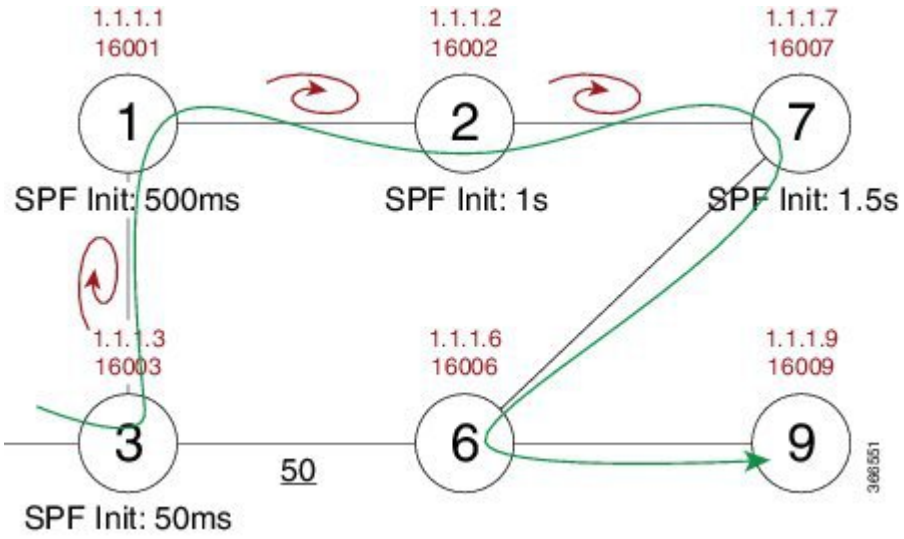
The following figure shows the Forwarding Information Base (FIB) table in each node before the link between Node 6 and Node 7 is established. The FIB entry contains the prefix of the destination node (Node 9) and the next hop.

Figure 22: Microloop Example Topology—FIB Entry



When the link between Node 6 and Node 7 comes up, microloops occur for the links based on the order of convergence of each node. In this example, Node 3 converges first with Node 1 resulting in a microloop between Node 3 and Node 1. Then, Node 1 converges next resulting in a microloop between Node 1 and Node 2. Next, Node 2 converges next resulting in a microloop between Node 2 and Node 7. Finally, Node 7 converges resolving the microloop and the packet reaches the destination Node 9, as shown in the following figure.

Figure 23: Microloop Example Topology—Microloops



Adding the SPF convergence delay, microloop results in a loss of connectivity for 1.5 seconds, which is the convergence duration specified for node 7.

Segment Routing and Microloops

The ISIS - SR: uLoop Avoidance feature supports the following scenarios:

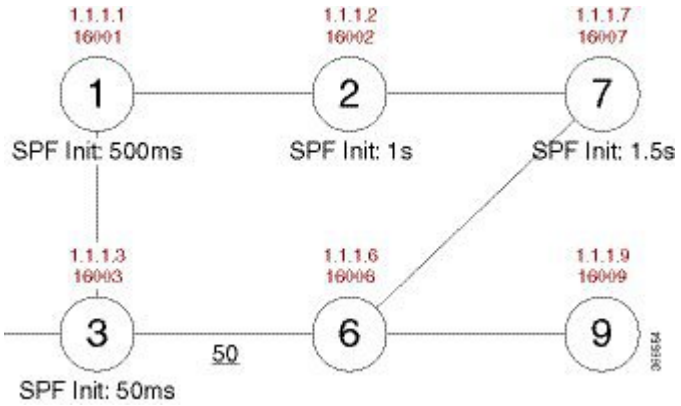
- Link-up or link-down for point-to-point links and a LAN segment with two nodes
- Link cost decrease or increase when a node is up or down due to the overload bit being set or unset

The **microloop avoidance segment-routing** command must be enabled on a node to prevent microloops.

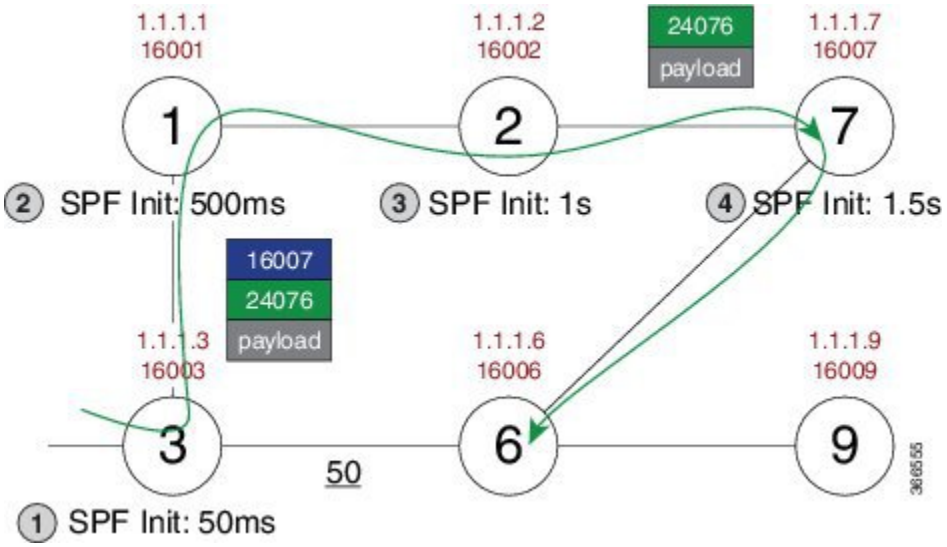
How Segment Routing Prevents Microloops?

Using the example used to explain microloops, this section explains how to segment routing prevents microloops. Node 3 in the example is enabled with the **microloop avoidance segment-routing** command. After the link between Node 6 and Node 7 comes up, Node 3 computes a new microloop on the network.

Figure 24: Microloop Example Topology—Segment Routing



Instead of updating the FIB table, Node 3 builds a dynamic loop-free alternate (LFA) SR TE policy for the destination (Node 9) using a list of segments IDs, which include the prefix segment ID (SID) of Node 7, which is 16007, and the adjacency segment ID (SID) of Node 6, which is 24076.



So, the SR TE policy enables a packet from Node 3 reaches its destination Node 9, without the risk of microloop until the network converges. Finally, Node 3 updates the FIB for the new path.

Use the protected keyword with the **microloop avoidance segment-routing** command, to enable microloop avoidance for protected prefixes only. The **microloop avoidance rib-update-delay** *milliseconds* command can be used to configure the delay in milliseconds for a node to wait before updating the node's forwarding table and stop using the microloop avoidance policy. The default value for the RIB delay is 5000 milliseconds.

How to Enable ISIS - SR: uLoop Avoidance

Enabling Microloop Avoidance

The following is a sample configuration code snippet to enable microloop avoidance.

```
router isis
 fast-reroute per-prefix level-2 all
 microloop avoidance segment-routing
 microloop avoidance rib-update-delay 3000
```

Verifying Microloop Avoidance

Use the **show isis rib** and **show ip route** commands to check if the repair path exists or not.

```
Router# show isis rib 10.20.20.0 255.255.255.0

IPv4 local RIB for IS-IS process sr

IPV4 unicast topology base (TID 0, TOPOID 0x0) ==================
Repair path attributes:
 DS - Downstream, LC - Linecard-Disjoint, NP - Node-Protecting
 PP - Primary-Path, SR - SRLG-Disjoint
```

```
10.20.20.0/24 prefix attr X:0 R:0 N:0 prefix SID index 2 - Bound (ULOOP EP)
 [115/L2/130] via 10.77.77.77(MPLS-SR-Tunnel5), from 10.44.44.44, tag 0,
LSP[2/5/29]
 prefix attr: X:0 R:0 N:0
 SRGB: 16000, range: 8000 prefix-SID index: None
 (ULOOP_EP)(installed)
 - - - - - -
 [115/L2/130] via 10.16.16.6(Ethernet2/0), from 10.44.44.44, tag 0, LSP[2/5/29]
 prefix attr: X:0 R:0 N:0
 SRGB: 16000, range: 8000 prefix-SID index: None
 (ALT)
```

```
Router# show ip route 10.20.20.0

Routing entry for 10.20.20.0/24
 Known via "isis", distance 115, metric 130, type level-2
 Redistributing via isis sr
 Last update from 10.77.77.77 on MPLS-SR-Tunnel5, 00:00:43 ago
 SR Incoming Label: 16002 via SRMS
 Routing Descriptor Blocks:
 * 10.77.77.77, from 10.44.44.44, 00:00:43 ago, via MPLS-SR-Tunnel5,
 * prefer-non-rib-labels, merge-labels
 Route metric is 130, traffic share count is 1
 MPLS label: 16002
 MPLS Flags: NSF
```

Additional References for ISIS - SR: uLoop Avoidance

Related Documents

Related Topic	Document Title
Segment Routing and IS-IS	*Using Segment Routing with IS-IS*
Overview of IS-IS concepts	"IS-IS Overview and Basic Configuration" module in the *IP Routing: ISIS Configuration Guide*
ISIS Local Microloop Protection	"ISIS Local Microloop Protection" module in the *IP Routing: ISIS Configuration Guide*

Standards/RFCs

Standard/RFC	Title
draft-francois-rtgwg-segment-routing-uloop-00	*Loop avoidance using Segment Routing*

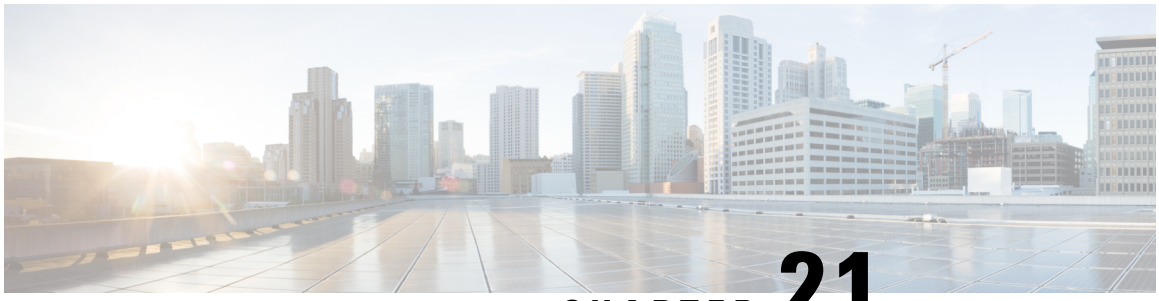
Feature Information for ISIS - SR: uLoop Avoidance

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 19: Feature Information for ISIS - SR: uLoop Avoidance

Feature Name	Releases	Feature Information
ISIS - SR: uLoop Avoidance	Cisco IOS XE Amsterdam 17.3.2	The ISIS - SR: uLoop Avoidance feature extends the ISIS Local Microloop Protection feature thereby preventing the occurrences of microloops during network convergence after a link-down event or link-up event. The following commands were introduced or modified: **microloop avoidance**, **microloop avoidance rib-update-delay**, **show mpls traffic tunnel**.



CHAPTER 21

BGP - SR: BGP Prefix SID Redistribution

The BGP - SR: BGP Prefix SID Redistribution feature provides support for BGP Prefix-SID in IPv4 prefixes in segment routing—BGP networks.

- Prerequisites for BGP - SR: BGP Prefix SID Redistribution, on page 197
- Information About BGP - SR: BGP Prefix SID Redistribution, on page 197
- How to Enable BGP - SR: BGP Prefix SID Redistribution, on page 198
- Additional References for BGP - SR: BGP Prefix SID Redistribution, on page 200
- Feature Information for BGP - SR: BGP Prefix SID Redistribution, on page 200

Prerequisites for BGP - SR: BGP Prefix SID Redistribution

- Mulitprotocol Label Switching (MPLS) must be configured.

Information About BGP - SR: BGP Prefix SID Redistribution

Segment Routing and BGP

Segment Routing uses Multiprotocol Label Switching (MPLS) labels to create a path to guide a packet in a network. Using segment routing, an MPLS label range is reserved with MPLS Forwarding Infrastructure (MFI). This label range is called Segment Routing Global Block (SRGB). A prefix SID assigned to a prefix is an extension of SRGB.

To support segment routing, Border Gateway Protocol (BGP) requires the ability to advertise a segment identifier (SID) for a BGP prefix. A BGP-Prefix-SID is the segment identifier of the BGP prefix segment in an segment routing with BGP network. A BGP-Prefix-SID is also an instruction to forward the packet over an ECMP-aware best-path computed by BGP to a related prefix. When BGP nodes communicate with neighbor nodes in a network, the BGP Update, message sent to neighbor nodes, includes the Prefix-SID Label in Labeled Unicast NLRI and a prefix SID index in a new attribute called Prefix SID attribute.

To support forwarding paths for traffic engineering, the forwarding path may need to be different from the optimal path. Hence, each BGP node assigns a local label to the neighbors and advertises the local label as adjacency SID through BGP--link state updates.

The BGP - SR: BGP Prefix SID Redistribution feature can be enabled by using the **connected-prefix-sid-map** command in the segment routing MPLS configuration mode. Additionally, you also need to enable the **segment-routing mpls** command in the router configuration mode for each address family.

Note In Cisco IOS XE Everest 16.6.1, IPv4 prefixes only are supported.

Segment Routing for Locally Sourced Routes

Interface host routes configured on local nodes are known as locally sourced routes. If segment routing is enabled, a BGP node includes the explicit or implicit null as prefix SID label and prefix SID attribute and advertises the prefix to a neighbor node.

If explicit-null is not configured on a neighbor, the MPLS Implicit Null label (3) is advertised to a neighbor node. If explicit-null is configured on a neighbor, the MPLS Explicit Null label corresponding to the address family of the prefix is advertised (0 for IPv4) to a neighbor node.

Segment Routing for Received Prefixes

BGP nodes that receive prefix SID attribute from a neighbor node via communication, add the label in the outgoing label as the prefix when a route is added to the RIB. The local label and prefix SID index is included in the RIB.

Segment Routing for Redistributed Routes

A source protocol on a BGP node allocates local label depending on the received prefix SID index and SRGB available on a local node. A source protocol provides the prefix SID index and the derived local label to RIB. BGP uses the local label from RIB as a label in the Labeled Unicast update sent to neighbors nodes.

BGP--MFI Interaction

BGP registers with MFI as a client and binds the label derived from SID index and SRGB as local label (with which traffic is expected to arrive) for the prefix.

How to Enable BGP - SR: BGP Prefix SID Redistribution

Enabling BGP-Prefix-SID

```
segment-routing mpls
 connected-prefix-sid-map */--------> Configures Prefix to SIDIndex Map that can be queried
 by BGP/IGP /*
 address-family ipv4
 10.0.0.1/255.0.0.0 index 10 range 10.11.0.1
```

Enabling BGP for Segment Routing

```
router bgp 2
 address-family-ipv4
  segment-routing mpls
```

Verifying BGP - SR: BGP Prefix SID Redistribution

This section shows how to verify the BGP - SR: BGP Prefix SID Redistribution feature with the help of an example network, in which, a device configured with segment routing is connected to two devices configured with Border Gateway Protocol (BGP). In each device, the **show segment-routing mpls** command is used to view the configuration.

The following is configuration on the device configured with segment routing.

```
segment-routing mpls
global-block 10000 13000
!
connected-prefix-sid-map
  address-family ipv4
   10.12.1.1/32 index 3 range 1
  exit-address-family
!
  segment-routing mpls


interface Loopback0
ip address 10.12.1.1 255.255.255.255

router bgp 1
neighbor 10.1.1.2 remote-as 2
!
address-family ipv4
  redistribute connected
  segment-routing mpls
  neighbor 10.1.1.2 activate
  neighbor 10.1.1.2 send-label
exit-address-family
```

The following is the configuration on the first device configured with BGP.

```
segment-routing mpls

router bgp 2
neighbor 10.1.1.1 remote-as 1
neighbor 10.11.1.2 remote-as 3
!
address-family ipv4
  redistribute connected
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 send-label
  neighbor 10.11.1.2 activate
  neighbor 10.11.1.2 send-label
exit-address-family
```

The following is the configuration on the second device configured with BGP.

```
segment-routing mpls

router bgp 3
neighbor 10.11.1.1 remote-as 2
!
address-family ipv4
```

```
  redistribute connected
  neighbor 10.11.1.1 activate
  neighbor 10.11.1.1 send-label
exit-address-family
```

Additional References for BGP - SR: BGP Prefix SID Redistribution

Related Documents

Standards and RFCs

Standard/RFC	Title
RFC3107	*Carrying Label Information in BGP-4*

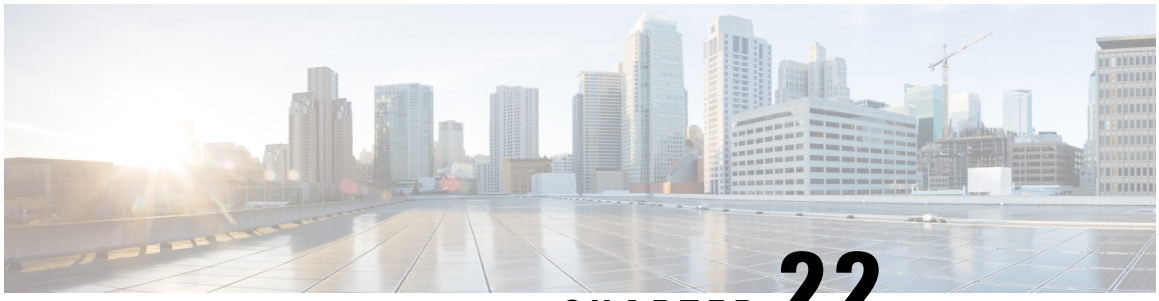
Feature Information for BGP - SR: BGP Prefix SID Redistribution

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 20: Feature Information for BGP - SR: BGP Prefix SID Redistribution

Feature Name	Releases	Feature Information
BGP - SR: BGP Prefix SID Redistribution	Cisco IOS XE Amsterdam 17.3.2	The BGP - SR: BGP Prefix SID Redistribution feature provides support for BGP Prefix-SID in IPv4 prefixes in segment routing—BGP networks. The following commands were introduced or modified: **connected-prefix-sid-map**, **segment-routing**.



CHAPTER 22

Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS

In a Segment Routing (SR) enabled network a centralized controller that programs SR tunnels needs to know the Maximum Segment Identifier (SID) Depth (MSD) supported by the head-end at node and/or link granularity to push the SID stack of an appropriate depth. MSD is relevant to the head-end of a SR tunnel or binding-SID anchor node where binding-SID expansions might result in creation of a new SID stack.

- Restrictions for Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS, on page 201
- Information About Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS, on page 201
- Verifying Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS , on page 203
- Feature Information for Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS, on page 204

Restrictions for Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS

- In IOS-XE as there no line cards, link-MSD is not advertised.

Information About Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS

Note This feature is enabled by default and no specific configuration is required to enable this functionality.

Maximum SID Depth

You can use IGPs to signal the MSD of a node or link to a centralized controller by:

- Advertising node-MSD to its peers.
- Providing the MSD information to BGP-LS.

Path Computation Element Protocol (PCEP) SR extensions signal MSD in SR PCE capability TLV and metric object. However, if PCEP is not supported/configured on the head-end of a SR tunnel or a binding-SID anchor node and controller does not participate in IGP routing, it has no way to learn the MSD of nodes. BGP-LS defines a way to expose topology and associated attributes and capabilities of the nodes in that topology to a centralized controller. Typically, BGP-LS is configured on a small number of nodes that do not necessarily act as head-ends. In order for BGP-LS to signal MSD for all the SR capable nodes in the network, MSD capabilities should be advertised by every IGP router in the network.

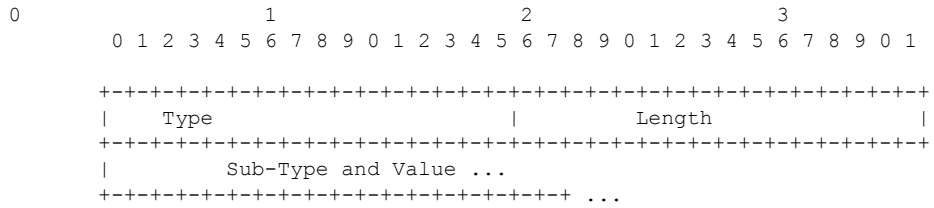
Readable Label Depth Capability (RLDC) is used by a head-end to insert Entropy Label (EL) at appropriate depth, so it could be read by transit nodes. MSD in contrary signals ability to push SID's stack of a particular depth.

MSD of type 1 (IANA registry) is used to signal the number of SIDs a node is capable of imposing to be used by a path computation element/controller. It is only relevant to the part of the stack created as the result of the computation. MSD advertises the total number of labels that a node is capable of imposing regardless of the number of service labels.

Node Maximum SID Depth Advertisement

A new Type/Length/Value (TLV) within the body called node MSD TLV is defined to carry the provisioned SID depth of the router originating the Router Information (RI) Link State Advertisement (LSA). Node MSD is the lowest MSD supported by the node.

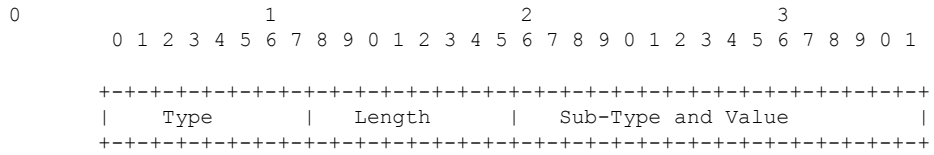
Node Maximum SID Depth Advertisement for OSPF



The Type (2 bytes) of this TLV is 12 (that is the suggested value to be assigned by IANA). Length is variable (minimum of 2, multiple of 2 octets) and represents the total length of value field. Value field consists of a 1 octet sub-type (IANA Registry) and 1 octet value.

Sub-Type 1, MSD and the value field contains maximum MSD of the device originating the RI LSA. Node maximum MSD falls in the range of 0-254. 0 represents lack of the ability to push MSD of any depth; any other value represents that of the node. This value should represent the lowest value supported by node.

Node Maximum SID Depth Advertisement for IS-IS



Node MSD is a sub-TLV for TLV 242. The type of this sub-TLV is 23. Length is variable (minimum of 2, multiple of 2 octets).

Sub-Type 1, MSD and the value field contains maximum MSD of the device originating the RI LSA. Node maximum MSD falls in the range of 0-254. 0 represents lack of the ability to push MSD of any depth; any other value represents that of the node. This value should represent the lowest value supported by node.

Getting the Node MSD from Hardware

IS-IS and OSPF are updated about the maximum SID Depth for the node from the underlying hardware. Based on that IS-IS and OSPF update the value in its TLVs.

Advertising the MSD to BGP-LS

lGP sends the information to LSLIB to make the MSD information available to BGP-LS. It can be node MSD or link MSD information. You also need to configure **distribute linkstate** under IS-IS for MSD to work. Perform the following steps to configure distribute link-state:

```
Device# configure terminal
Device(config)# router isis
Device(config-router)# distribute link-state
```

Verifying Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS

Verifying Advertise Maximum SID Depth Using IS-IS

The following show command is used to verify the node MSD TLV:

```
Device# show isis database verbose
Router CAP:   10.10.10.1, D:0, S:0
    Segment Routing: I:1 V:0, SRGB Base: 16000 Range: 8000
       Segment Routing Algorithms: SPF, Strict-SPF
      Router CAP:   10.2.2.2, D:0, S:0
    Segment Routing: I:1 V:0, SRGB Base: 16000 Range: 8000
    Segment Routing Algorithms: SPF, Strict-SPF
    Node-MSD
      MSD: 16
```

Verifying Advertise Maximum SID Depth Using OSPF

The following show command is used to verify the node MSD TLV:

```
Device# show ip ospf database opaque-area type router-information
TLV Type: Segment Routing Node MSD
   Length: 2
     Sub-type: Node Max Sid Depth, Value: 16
```

Feature Information for Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 21: Feature Information for Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS

Feature Name	Releases	Feature Information
Advertise Maximum SID Depth by IS-IS and OSPF to BGP-LS	Cisco IOS XE Amsterdam 17.3.2	In a Segment Routing (SR) enabled network a centralized controller that programs SR tunnels needs to know the Maximum Segment Identifier (SID) Depth (MSD) supported by the head-end at node and/or link granularity to push the SID stack of an appropriate depth. MSD is relevant to the head-end of a SR tunnel or binding-SID anchor node where binding-SID expansions might result in creation of a new SID stack. The following commands were introduced or modified by this feature: **distribute link-state**, **show isis database verbose**, **show ip ospf database opaque-area type router-information**.



CHAPTER 23

RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel

This document describes support for link protection also referred as next-hop (NHOP) protection using the backup Segment-Routing Traffic Engineering (SR-TE) autotunnel. It protect the links over which the RSVP Traffic Engineering (RSVP-TE) tunnel traverses.

- Feature Information for RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel, on page 205
- Prerequisites for RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel, on page 206
- Restrictions for RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel, on page 206
- Information About RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel, on page 207
- How to Configure RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel, on page 209
- Verifying RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel, on page 211

Feature Information for RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 22: Feature Information for RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel

Feature Name	Releases	Feature Information
RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel	Cisco IOS XE Amsterdam 17.3.2	This feature provides support for link protection also referred as next-hop (NHOP) protection using the backup Segment-Routing Traffic Engineering (SR-TE) autotunnel. It protect the links over which the RSVP Traffic Engineering (RSVP-TE) tunnel traverses. The following commands were introduced by this feature: **ip explicit-path name path1 enable**, **show mpls traffic-eng tunnels tunnel 65436**, **show ip explicit-paths**, **show mpls traffic-eng tunnels tunnel 65436 | show Segment-Routing Path Info**, **show mpls traffic-eng fast-reroute database**, **show ip rsvp fast-reroute sh mpls traffic-eng auto-tunnel backup**.

Prerequisites for RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel

Before enabling SR-TE backup autotunnel, ensure that the following technologies are configured in your setup:

- IS-IS Network Point to Point Interfaces
- Segment Routing

Additionally, prior knowledge of the following technologies are required:

- MPLS Traffic-Engineering
- RSVP Traffic-Engineering
- Fast reroute

Restrictions for RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel

- SR-TE backup autotunnel cannot be used for bandwidth protection.
- SR-TE backup autotunnel can only be used as a backup for RSVP-TE tunnel protection.

Information About RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel

Benefits of RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel

With increased complexity in the network, scalability becomes a challenge due to maintenance of RSVP-TE tunnels with complex signaling as well as high overhead on routers within the network. Backup autotunnel feature can help reduce the complexity in a Segment Routing (SR) network. Autotunnel backup feature has the following benefits:

- Backup tunnels are built automatically hence eliminating the need for users to pre-configure each backup tunnel and then assign the backup tunnel to the protected interface.
- With the backup tunnels configured, area of protection gets expanded. Fast reroute (FRR) neither protects IP traffic nor LDP labels that do not use TE tunnel.
- Backup SR-TE autotunnel allows additional means of migration to SR network without disrupting the existing traffic passing through RSVP-TE tunnels.

Backup AutoTunnel

Backup autotunnels on a router helps to build dynamic backup tunnels whenever required. This prevents creating of static SR-TE tunnels.

To protect a label-switched path (LSP) in the absence of static SR-TE tunnels, you need to do the following:

- Preconfigure each backup tunnel.
- Assign the backup tunnels to the protected interfaces.

An LSP requests backup protection from Resource Reservation Protocol (RSVP) FRR in the following situations:

- Receipt of the first RSVP Resv message.
- Receipt of an RSVP path message with the protection attribute after the LSP has been established without protection attribute.
- Detection of changed Record Route Object (RRO).

If there is no backup tunnel protecting the interface used by the LSP, the LSP remained unprotected. Some of the reasons why a backup tunnel may not be available are:

- Static backup tunnels are not configured.
- Static backup tunnels are configured, but may not be able to protect the LSP because there is not enough bandwidth available, or the tunnel protects a different pool, or the tunnel is not available.

If a backup tunnel is not available, the following two backup tunnels are created dynamically:

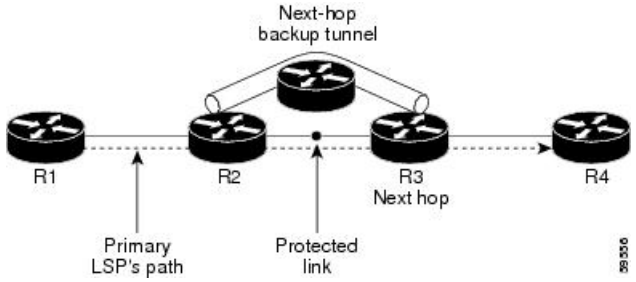
- NHOP—Protects against link failure.
- NNHOP—Protects against node failure.

Note At the penultimate hop, only an NHOP backup tunnel is created.

Link Protection

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 are referred to as NHOP backup tunnels because they terminate at the LSP's next hop beyond the point of failure.

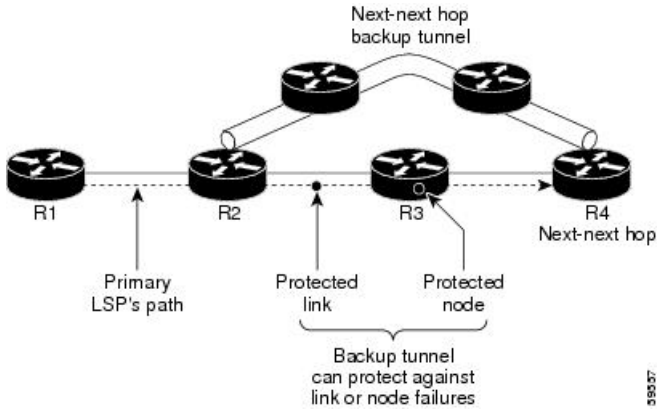
Figure 25: Next-Hop Backup Tunnel



Node Protection

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

Figure 26: Next-Next Hop Backup Tunnel



Explicit Paths

Explicit paths are used to create backup autotunnels as follows:

- NHOP excludes the protected link's IP address.
- NNHOP excludes the NHOP router ID.
- The explicit-path name is _auto-tunnel_tunnel *xxx*, where *xxx* matches the dynamically created backup tunnel ID.

Range for Backup AutoTunnels

You can configure the tunnel range for backup autotunnels. By default, the last 100 TE tunnel IDs are used, which is 65,436 to 65,535. Autotunnels detect tunnel IDs that are alloted starting with the lowest number.

For example, if you configure a tunnel within the range of 1000 to 1100. And statically configured TE tunnel also falls in the same range then routers do not use those IDs. If those static tunnels are removed, the MPLS-TE dynamic tunnel software can use those IDs.

How to Configure RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel

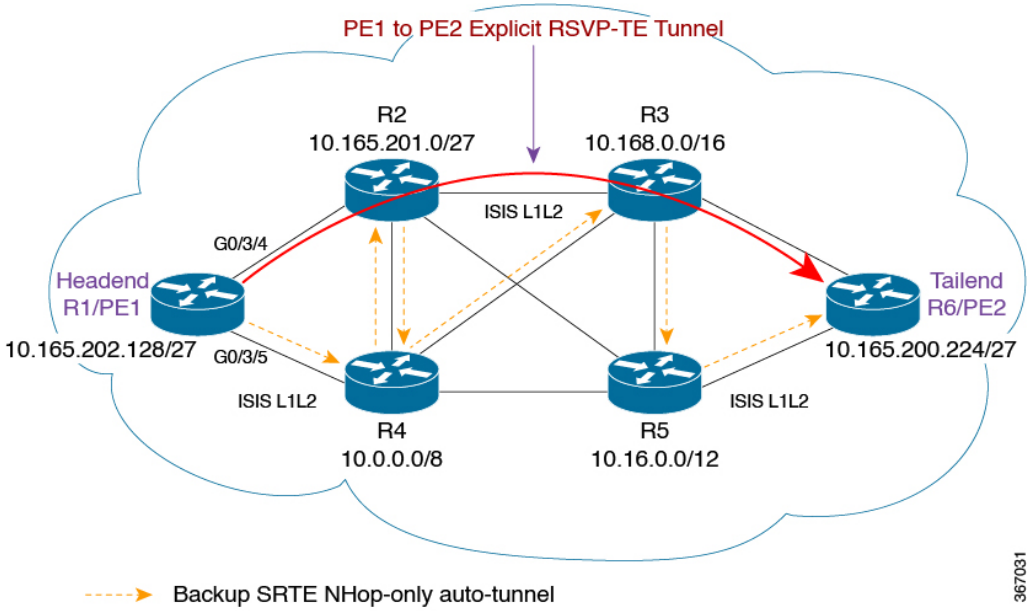
Configuring Explicit Path for Point-to-Point Network Type

For SR-TE autotunnel backup feature to work interfaces have to be point-to-point network type.

```
interface Loopback0
 ip address 10.51.1.1 255.255.255.255
 ip router isis 1
end
!
interface GigabitEthernet0/2/0
 ip address 10.102.6.1 255.255.255.0
 ip router isis 1
 negotiation auto
 mpls traffic-eng tunnels
 isis network point-to-point
 ip rsvp bandwidth
end
!
interface GigabitEthernet0/2/4
 ip address 10.104.1.1 255.255.255.0
 ip router isis 1
 negotiation auto
 mpls traffic-eng tunnels
 isis network point-to-point
 ip rsvp bandwidth
end
```

Configuring Explicit RSVP-TE Tunnel With FRR

Figure 27: Explicit RSVP-TE Tunnel



1. Configure explicit path from R1/PE1 to R6/PE2 that traverses through the routers R2 and R3.

```
ip explicit-path name path1 enable
 index 1 next-address 10.165.202.128
 index 2 next-address 10.165.201.0
 index 3 next-address 10.168.0.0
 index 4 next-address 10.165.200.224
```

2. Configure explicit RSVP-TE tunnel.

```
interface Tunnel1
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 10.165.200.224
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 10 explicit name path1
 tunnel mpls traffic-eng record-route
end
```

3. Configure the primary RSVP-TE Tunnel 1 with FRR to activate the protection process.

```
interface tunnel 1
 tunnel mpls traffic-eng fast-reroute
```

4. Configure the global command to enable link protection using SR-TE autotunnel.

```
mpls traffic-eng auto-tunnel backup segment-routing nhop-only
```

Note This command needs to be available in all the nodes that require link protection.

The Primary RSVP-TE tunnel need to be protected that gets initialized from headend R1/PE1 to destination R6/PE2 and traversing through next node R2 and so on. In this case, R1/PE1 is the Point of Local Repair (PLR) and R2 is the Mid-Point (MP). With link protection, the SR-TE Backup AutoTunnel provides protection to the link from R1/PE1 to R2 by traversing through the path R1/PE1 -> R4 and R4 -> R2, hence converging back to the MP.

Verifying RSVP-TE Protection using Segment Routing Traffic Engineering AutoTunnel

Use the **show interfaces Tunnel** command to verify if SR-TE AutoTunnel is generated and up.

```
Device#show interfaces Tunnel65436
Tunnel65436 is up, line protocol is up
```

Use the **show mpls traffic-eng tunnels** command to verify if the backup AutoTunnel is a SR-TE Tunnel.

```
Device#show mpls traffic-eng tunnels tunnel 65436
Name: R1_t65436 (Tunnel65436) Destination: 10.165.201.0
 Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path option 1, (SEGMENT-ROUTING) type explicit __dynamic_tunnel65436 (Basis for
Setup, path weight 20)
```

Use the **show ip explicit-paths** command to verify if the SR-TE Backup Tunnel is using a secondary path to reach the node.

```
Device#show ip explicit-paths
PATH __dynamic_tunnel65436 (strict source route, path complete, generation 49, status
non-configured)
1: exclude-address 10.102.5.1
```

Use the **show mpls traffic-eng tunnels tunnel 65436 | s Segment-Routing Path Info** command to verify if the backup tunnel is going through the path R1/PE1 to R4 and finally to destination R2 which is the mid-point.

```
Device#show mpls traffic-eng tunnels tunnel 65436 | s Segment-Routing Path Info
Segment-Routing Path Info (isis level-1)
 Segment0[Link]: 10.104.1.1 - 10.104.1.2, Label: 19
 Segment1[Link]: 10.104.6.2 - 10.104.6.1, Label: 18
```

Use the **show mpls traffic-eng auto-tunnel backup** command to verify if the auto-tunnel backup state is correct.

```
Device#show mpls traffic-eng auto-tunnel backup
State: Enabled
 Auto backup tunnels: 1 (up: 1, down: 0)
 Tunnel ID Range: 65436 - 65535
```

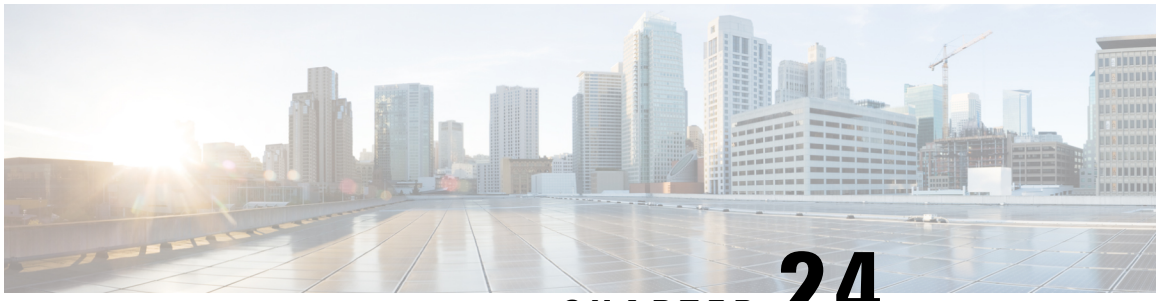
```
 Create Nhop Only: Yes
 Check for deletion of unused tunnels every: 3600 Sec
 SRLG: Not configured

Config:
 unnumbered-interface: Loopback0
  Affinity/Mask: 0x0/0xFFFF
```

Use the **show mpls traffic-eng fast-reroute database** command to verify if the primary link through which the RSVP-TE LSP is traversing is protected.

```
Device#show mpls traffic-eng fast-reroute database
P2P Headend FRR information:
Protected tunnel In-label Out intf/label FRR intf/label Status
-------------------------- -------- ------------- ------------- ------
Tunnel1 Tun hd Gi0/3/4:30 Tu65436:30 ready


Device#show ip rsvp fast-reroute
P2P Protect BW Backup
Protected LSP I/F BPS:Type Tunnel:Label State Level Type
------------- ------- -------- ------------- ------ ----- ------
R1_t1 Gi0/3/4 0:G Tu65436:28 Ready any-unl Nhop
```



CHAPTER 24

ISIS Manual Adjacency SID

The Integrated Intermediate System-to-Intermediate System (IS-IS) manual adjacency SID feature provides information about manually provisioned Adjency SIDs.

- Feature Information for ISIS Manual Adjacency SID, on page 213
- Information About ISIS Manual Adjacency SID, on page 213
- Configuring Manual Adjacency SID, on page 215
- Verifying Manual Adjacency SID, on page 216

Feature Information for ISIS Manual Adjacency SID

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 23: Feature Information for ISIS Manual Adjacency SID

Feature Name	Releases	Feature Information
ISIS Manual Adjacency SID	Cisco IOS XE Amsterdam 17.3.2	The Integrated Intermediate System-to-Intermediate System (IS-IS) manual adjacency SID feature provides information about manually provisioned Adjency SIDs. The following commands were introduced by this feature: **adjacency-sid [absolute | index]**<*value*> **[protected]**. .

Information About ISIS Manual Adjacency SID

Segment routing (SR) networks often use SR Traffic Engineering (SR-TE) to influence the path the specific traffic takes over the network. SR-TE tunnels can be provisioned manually on the tunnel head, but often they are calculated and provisioned by the central controller. In many cases operator of the network wants to be able to force the traffic over specific nodes and links.

To force the traffic over a certain node in the SR network operators can use Prefix-SID that is advertised by the node. Many times the anycast Prefix SID is used which forces the traffic to go over specific location where multiple nodes share the same Prefix-SID.

To force the traffic over the specific link, an Adjacency-SID (Adj-SID) is used. The problem with the existing implementation of the Adj-SID is that it is a dynamically allocated value which is in contrast to manually provisioned prefix-SID. The fact that the Adj-SID is dynamically allocated brings a set of problems:

- The value is not persistent over reload or process restart.
- The value is not known upfront so controller cannot use it unless it has access to the information flooded by IGP (natively or through BGP-LS).
- Each link is allocated a unique adj-SID value which prevents the same adj-SID to be shared by multiple links.

To address the above mentioned issues, the adj-SIDs are enhanced and now thay are capable of the following:

- Support manually provisioned adj-SID that is persistent over reload and restart.
- Support same adj-SID to be provisioned for multiple adjacencies to the same neighbor.
- Support same adj-SID to be provisioned for multiple adjacencies going to different neighbors.
- Multiple manual Adj-SIDs can be configured for a single adjacency.

Manual Adjacency SID

The existing IS-IS Adj-SID infrastructure that is being used for dynamically allocated Adj-SIDs is extended to support the new persistent Adj-SID requirements. A new CLI command is also introduced to manually assign Adj-SID values for point-to-point links. Multiple Adj-SIDs can be provisioned on a single point-to-point interface. Same Adj-SID can be provisioned on multiple point-to-point interfaces leading to the same or different neighbors.

All manual Adj-SIDs are assigned from a range of labels called Segment Routing Local Block(SRLB). The default SRLB Range is 15000-15999.

Manual Adj-SIDs can be configured as an Index or an Absolute value. If it is configured as an index, the absolute label is calculated as an index + SRLB starting label. For example, if you configure 56 as a manual Adj-SID index, the absolute label would be 15000 + 56 = 15056. If it is configured as an absolute, the label itself is the absolute value. For example, if you configure 56 as an absolute manual Adj-SID, the absolute label would be 56 only. Labels (both index and absolute) can be configured as protected or non-protected. By default, all the labels are non-protected.

Adjacency SID Advertisement

Manual adj-SIDs are advertised using existing ISIS adj-SID sub-TLV as defined in the ISIS SR extension draft. If the same value of the adj-SID has been provisioned on multiple interfaces, the S-Flag is set in the adj-SID sub-TLV. In the case of manual adj-SID, P flag is always set.

If the provisioned adj-SID has been configured as protected, the B-flag also gets set.

Adjacency-SIDs are always advertised as a label value and never as an index even if the index are used to configure the adj-SID.

Adjacency SID Forwarding

When the adj-SID value is only configured on a single interface, then the ISIS installs forwarding entries for manually allocated adj-SIDs. The primary path for any Adj-SID is a POP operation over the point-to-point interface for which the Adj-SID is allocated. If the allocated adj-SID is eligible for backup and the backup path is available, IS-IS programs the backup path as well. The backup path for Adj-SID is equal to the backup path computed for the neighbor router-id address.

If the same adj-SID value is configured on multiple links forwarding happens as the following:

- Primary path with POP operation is installed via each link where adj-SID is configured with that value.
- For each primary path if the adj-SID is configured as protected on the primary interface and backup is available, backup path gets installed. Backup path is represented as a backup path associated with the neighbor router-id address.

Configuration Prerequisites

- Ensure that segment routing is configured globally.
- Ensure that segment routing is configured using IS-IS.

Configuring Manual Adjacency SID

```
Device#configure terminal
Device(config)#interface ethernet0/1
Device(config-if)#isis adjacency-sid [absolute | index] <value> [protected]
```

[index] – (Optional) It is used if the adjacency SID is configured as an index to the SRLB range. If the index keyword is not used the value is expected to represent the absolute value of the label.

[absolute] - (Optional) It is used if the adjacency SID is configured as absolute value.

<*value*> - It represents the adj-SID label value or index. For the adj-SID to be programmed and advertised, the value/index must fall in the valid SRLB range.

[protected] - (Optional) It is used to protect the manual adj-SIDs. By default, manual adj-SIDs are not protected.

Modifying Segment Routing Local Block (SRLB) Range

```
Device#configure terminal
Device(config)#segment-routing mpls
Device(config-srmpls)#local-block 7000 7999
```

Verifying Manual Adjacency SID

Verifying Label in SR APP Database

```
Device#show segment-routing mpls lb assigned-sids
Adjacency SID Database
 C=> In conflict
 S=> Shared
 R=> In range
SID  STATE       PROTOCOL     TOPOID      LAN      PRO NEIGHBOR    INTERFACE
15378 R
                 ISIS         0           N        N   10.0.0.3     Ethernet0/1
```

Verifying Label in MPLS Forwarding

```
       Device# show mpls forwarding-table
       Local      Outgoing        Prefix               Bytes Label      Outgoing
Next Hop
       Label      Label           or Tunnel Id         Switched         interface
       15378      Pop Label       0.0.60.18-A          0                Et0/0
10.0.0.2 □== Configured only for interface e0/0
```

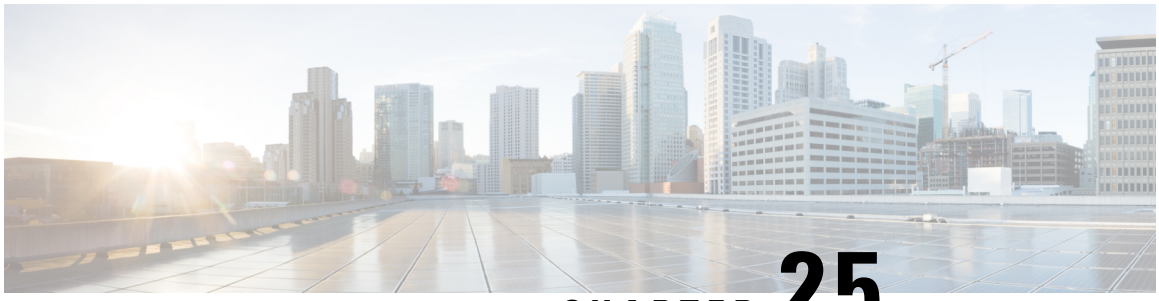
Verifying Shared Label

```
       Device# show mpls forwarding-table
       Local      Outgoing       Prefix               Bytes Label      Outgoing        Next
 Hop
       Label      Label          or Tunnel Id         Switched         interface
       15378      Pop Label      0.0.60.18-A          0                Et0/0
10.0.0.2  □== Same Label is configured for 2 interfaces
                  Pop Label      0.0.60.18-A          0                Et0/1
10.0.0.3  □==
```

Verifying ISIS LSP

```
       Device# sh isis database verbose R1.00-00
       xxxxxx
       xxxxxx
       Adjacency SID Value:15378 F:0 B:0 V:1 L:1 S:1 P:1 Weight:0 □== P (Persistent) flag
is always 1 if it is Manual Adj-SID
       xxxxxx

       P -> Persistent Flag (0 for Dynamic Adj-SID and 1 for Manual Adj-SID)
       S -> Shared Flag (1 if label is shared by multiple adjacencies)
```



CHAPTER 25

OSPF Manual Adjacency SID

The OSPF manual adjacency SID feature supports configuration of static adjacency SIDs for Segment Routing with OSPFv2.

- Feature Information for OSPF Manual Adjacency SID, on page 217
- Information About OSPF Manual Adjacency SID, on page 217
- How to Configure OSPF Manual Adjacency SID, on page 219

Feature Information for OSPF Manual Adjacency SID

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 24: Feature Information for OSPF Manual Adjacency SID

Feature Name	Releases	Feature Information
OSPF Manual Adjacency SID	Cisco IOS XE Amsterdam 17.3.2	The OSPF manual adjacency SID feature supports configuration of static adjacency SIDs for Segment Routing with OSPFv2. The following command was introduced by this feature: **adjacency-sid index** *value* [**protected**]

Information About OSPF Manual Adjacency SID

Segment routing (SR) networks often use SR Traffic Engineering (SR-TE) to influence the path specific traffic takes over the network. SR-TE tunnels can be provisioned manually on the tunnel headend, or are calculated and provisioned by a central controller.

For traffic engineering, operators of a network need to be able to force traffic over specific nodes and links. To force traffic over a certain node in the SR network, operators can use the Prefix SID that is advertised by

the node. An anycast Prefix SID can be used to route traffic to specific node when multiple nodes advertise the same Prefix SID.

To force traffic through a certain link, operators can use the adjacency SID of the link. Without the support for manually-configured adjacency SIDs, adjacency SIDs are dynamically allocated. Dynamically allocated SIDs have the following disadvantages in relation to traffic engineering:

- The dynamic value is not persistent over reload or process restart.
- The dynamic value is not known upfront, and so a controller cannot use it unless it has access to the information flooded by IGP (natively or through BGP-LS).
- Each link is allocated a unique Adjacent SID value. With such an allocation, the same adjacency SID cannot be allocated to multiple links.

The OSPF Manual Adjacency SID feature introduces support for manually-configured adjacency SIDs. With manually-configured static adjacency SIDs,

- provisioned adjacency SID is persistent over reload and restart.
- multiple adjacency SIDs can be configured for a single adjacency.

Prerequisites for OSPF Manual Adjacency SID

- Segment Routing must be configured globally.
- Segment Routing must be configured for the OSPF instance.

Restrictions for OSPF Manual Adjacency SID

- Static adjacency SIDs can be configured only for point-to-point links and not for broadcast links.
- Do not assign the same adjacency SID to multiple links. Group adjacency SIDs are not supported.
- Do not configure the same static adjacency SID in multiple IGPs or IGP instances. Such a configuration is not supported and a conflict handling mechanism for the scenario is yet to be implemented.
- Specify static adjacency SIDs as an indices to the Segment Routing Local Block (SRLB). Static adjacency SIDs cannot be specified as absolute values of labels in the SRLB.

Manual Adjacency SIDs

Static adjacency SIDs can be configured for point-to-point links with OSPFv2.

Manual adjacency SIDs must be assigned from the SRLB. The default range of SRLB labels is 15000 to 15999. You can modify the SRLB range using the **local-block** *range-start range-end* command.

You can assign static adjacency SIDs as indices to the SRLB. Based on the index assigned, the label for the adjacency SID is calculated as `label = SRLB_range_start + index_value`.

By default, static adjacency SIDs are not protected, and therefore, you can specify whether a static adjacency SID must be protected or not during configuration.

Manual Adjacency SID Advertisement

Static adjacency SIDs are advertised using the existing Adj-SID Sub-TLV of the Extended Link LSA as defined in OSPF Extensions for Segment Routing.

For static adjacency SIDs, the P-flag (Persistent flag) is set in the Adj-SID Sub-TLV.

If a static adjacency SID is protected, then the B-flag is set in the Adj-SID Sub-TLV.

Static adjacency SIDs are always advertised as labels. When the static adjacency SID is configured as an index, the absolute value of the label is calculated and the label value is advertised.

Manual Adjacency SID Forwarding

When a static adjacency SID is configured for a point-to-point interface, OSPFv2 installs forwarding entries for the manually allocated adjacency SID. The primary path for an adjacency SID is a POP operation over the point-to-point interface for which the adjacency SID is allocated.

If the manually-allocated adjacency SID is eligible for backup and a backup path is available, OSPFv2 programs the backup path as well. The backup path for a manually-allocated adjacency SID is the backup path computed for the neighbor router.

How to Configure OSPF Manual Adjacency SID

Modifying Segment Routing Local Block Range

```
Device#configure terminal
Device(config)#segment-routing mpls
Device(config-srmpls)#local-block range-start range-end
```

range-start and *range-start* indicate the modified range bounds for Segment Routing Local Block (SRLB).

OSPF advertises the SRLB in the SR Local Block TLV of the Router Information (R.I.) Opaque LSA.

Only a single range is supported for SRLB. If an SR Local Block TLV has multiple ranges, the receiving router ignores the TLV.

```
Device#configure terminal
Device(config)#segment-routing mpls
Device(config-srmpls)#local-block 7000 7999
```

Configuring OSPF Manual Adjacency SID

```
Device#configure terminal
Device(config)#interface <interface>
Device(config-if)#ip ospf adjacency-sid index <sid_value> [protected]
```

`<sid_value>` must be an index to the SRLB. Configuration of the adjacency SID as an absolute label value is yet to be supported.

`[protected]` (Optional) – This keyword is used to protect a manual adjacency SID. By default, manual adjacency SIDs are not protected.

Verifying OSPF Manual Adjacency SID

You can verify SIDs assigned to adjacencies and whether an SID is static or dynamic using the commands **show ip ospf segment-routing adjacency-sid** and **show ip ospf segment-routing adjacency-sid detail**. The output of either command also shows additional information such as the neighbor linked through an adjacency, whether an adjacency is protected or not, and the backup next-hop and interface for a protected adjacency.

- `router#show ip ospf segment-routing adjacency-sid`

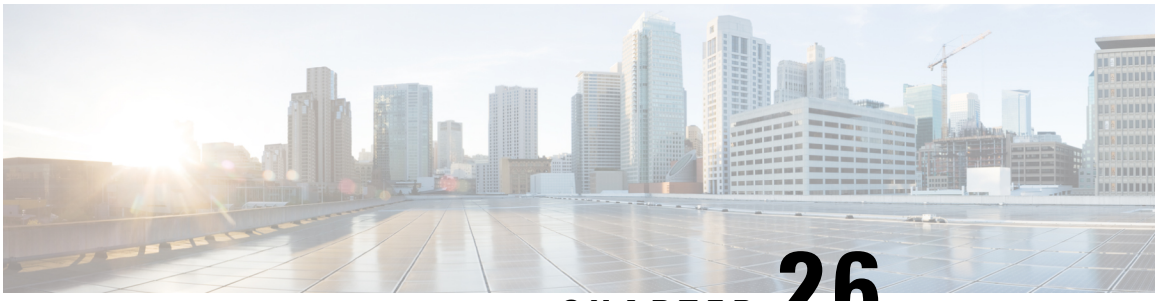
```
            OSPF Router with ID (10.2.0.0) (Process ID 1)
    Flags: S - Static, D - Dynamic,  P - Protected, U - Unprotected, G - Group, L -
Adjacency Lost

Adj-Sid  Neighbor ID     Interface          Neighbor Addr   Flags   Backup Nexthop
Backup Interface
-------- --------------- ------------------ --------------- ------- ---------------
------------------
16       10.3.0.0         Et0/2.3            10.3.3.3         D U
17       10.3.0.0         Et0/2.1            10.3.1.3         D U
24       10.3.0.0         Et0/2.1            10.3.1.3         D P     10.3.2.3
 Et0/2.2
25       10.1.0.0         Et0/0              10.2.0.1         D U
26       10.1.0.0         Et0/0              10.2.0.1         D P     10.3.1.3
 Et0/2.1
27       10.3.0.0         Et0/2.2            10.3.2.3         D U
28       10.3.0.0         Et0/2              10.3.0.3         D U
29       10.3.0.0         Et0/2              10.3.0.3         D P     10.4.0.4
 Et0/1
30       10.4.0.0         Et0/1              10.4.0.4         D U
34       10.4.0.0         Et0/1              10.4.0.4         D P     10.3.1.3
 Et0/2.1
15010    10.1.0.0         Et0/0              10.2.0.1         S P     10.3.1.3
 Et0/2.1
15210    10.1.0.0         Et0/0              10.2.0.1         S U
15230    10.3.0.0         Et0/2              10.3.0.3         S P     10.4.0.4
 Et0/1
15240    10.4.0.0         Et0/1              10.4.0.4         S U
15800    10.3.0.0         Et0/2.1            10.3.1.3         S U
15801    10.3.0.0         Et0/2.2            10.3.2.3         S U
15802    10.3.0.0         Et0/2.3            10.3.3.3         S U
15810    10.3.0.0         Et0/2.1            10.3.1.3         S P     10.3.2.3
 Et0/2.2
```

- `router#show ip ospf segment-routing adjacency-sid detail`

```
            OSPF Router with ID (10.2.0.0) (Process ID 1)
Label 16, Paths 1, Dynamic
   Nbr id 10.3.0.0, via 10.3.3.3 on Et0/2.3, Unprotected
Label 17, Paths 1, Dynamic
   Nbr id 10.3.0.0, via 10.3.1.3 on Et0/2.1, Unprotected
Label 24, Paths 1, Dynamic
   Nbr id 10.3.0.0, via 10.3.1.3 on Et0/2.1, Protected, Nbr Prefix 10.33.33.33
      Primary path: via 10.3.1.3 on Et0/2.1, out-label 3
      Repair path: via 10.3.2.3 on Et0/2.2, out-label 3, cost 31, labels 0
Label 25, Paths 1, Dynamic
   Nbr id 10.1.0.0, via 10.2.0.1 on Et0/0, Unprotected
Label 26, Paths 1, Dynamic
   Nbr id 10.1.0.0, via 10.2.0.1 on Et0/0, Protected, Nbr Prefix 10.1.1.1
      Primary path: via 10.2.0.1 on Et0/0, out-label 3
      Repair path: via 10.3.1.3 on Et0/2.1, out-label 16001, cost 31, labels 0
Label 27, Paths 1, Dynamic
   Nbr id 10.3.0.0, via 10.3.2.3 on Et0/2.2, Unprotected
Label 28, Paths 1, Dynamic
```

```
   Nbr id 10.3.0.0, via 10.3.0.3 on Et0/2, Unprotected
Label 29, Paths 1, Dynamic
   Nbr id 10.3.0.0, via 10.3.0.3 on Et0/2, Protected, Nbr Prefix 10.3.3.3
      Primary path: via 10.3.0.3 on Et0/2, out-label 3
      Repair path: via 10.4.0.4 on Et0/1, out-label 16003, cost 21, labels 0
Label 30, Paths 1, Dynamic
   Nbr id 10.4.0.0, via 10.4.0.4 on Et0/1, Unprotected
Label 34, Paths 1, Dynamic
   Nbr id 10.4.0.0, via 10.4.0.4 on Et0/1, Protected, Nbr Prefix 10.4.4.4
      Primary path: via 2.4.0.4 on Et0/1, out-label 3
      Repair path: via 10.3.1.3 on Et0/2.1, out-label 16004, cost 31, labels 0
Label 15010, Paths 1, Static
   Nbr id 10.1.0.0, via 10.2.0.1 on Et0/0, Protected, Nbr Prefix 10.1.1.1
      Primary path: via 10.2.0.1 on Et0/0, out-label 3
      Repair path: via 10.3.1.3 on Et0/2.1, out-label 16001, cost 31, labels 0
Label 15210, Paths 1, Static
   Nbr id 10.1.0.0, via 10.2.0.1 on Et0/0, Unprotected
Label 15230, Paths 1, Static
   Nbr id 10.3.0.0, via 10.3.0.3 on Et0/2, Protected, Nbr Prefix 10.3.3.3
      Primary path: via 10.3.0.3 on Et0/2, out-label 3
      Repair path: via 10.4.0.4 on Et0/1, out-label 16003, cost 21, labels 0
Label 15240, Paths 1, Static
   Nbr id 10.4.0.0, via 10.4.0.4 on Et0/1, Unprotected
Label 15800, Paths 1, Static
   Nbr id 10.3.0.0, via 10.3.1.3 on Et0/2.1, Unprotected
Label 15801, Paths 1, Static
   Nbr id 10.3.0.0, via 10.3.2.3 on Et0/2.2, Unprotected
Label 15802, Paths 1, Static
   Nbr id 10.3.0.0, via 10.3.3.3 on Et0/2.3, Unprotected
Label 15810, Paths 1, Static
   Nbr id 10.3.0.0, via 10.3.1.3 on Et0/2.1, Protected, Nbr Prefix 10.33.33.33
      Primary path: via 10.3.1.3 on Et0/2.1, out-label 3
      Repair path: via 10.3.2.3 on Et0/2.2, out-label 3, cost 31, labels 0
```



CHAPTER 26

OSPFv2 Segment Routing Strict SPF

The OSPFv2 Segment Routing Strict Shortest Path First (SPF) feature provides information about the strict SPF segment identifiers (SIDs).

- Feature Information for OSPFv2 Segment Routing Strict SPF, on page 223
- Restrictions for OSPFv2 Segment Routing Strict SPF, on page 224
- Information About OSPFv2 Segment Routing Strict SPF, on page 224
- Enabling and Disabling OSPFv2 Segment Routing Strict SPF, on page 225
- Configuring OSPFv2 Segment Routing Strict SPF SID, on page 226
- Verifying OSPFv2 Segment Routing Strict SPF, on page 226

Feature Information for OSPFv2 Segment Routing Strict SPF

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 25: Feature Information for OSPFv2 Segment Routing Strict SPF

Feature Name	Releases	Feature Information
OSPFv2 Segment Routing Strict SPF	Cisco IOS XE Amsterdam 17.3.2	The OSPFv2 Segment Routing Strict SPF feature provides the provision to support strict shortest path algorithm. It mandates that the packets are forwarded according to SPF algorithm and instructs any router in the path to ignore any possible local policy overriding the SPF decision. The following commands were added or modified: **address-family ipv4 strict-spf**.

Restrictions for OSPFv2 Segment Routing Strict SPF

- All the nodes in an OSPF area must be strict SPF capable and each node must have at least one strict SPF SID for the strict SPF solution to work with segment routing traffic engineering (SR-TE).
- Redistribution of strict SPF sid is not supported.

Information About OSPFv2 Segment Routing Strict SPF

Segment Routing (SR) architecture provides the provision to support multiple prefix-SID algorithms. Currently, it has defined two algorithms:

- **Algorithm 0** – This is a shortest path algorithm and it is supported by default.
- **Algorithm 1** – This is a strict shortest path algorithm. It mandates that the packets are forwarded according to SPF algorithm and instructs any router in the path to ignore any possible local policy overriding the SPF decision. The SID advertised with strict shortest path algorithm ensures that the path the packet is going to take is the expected path, and not the altered SPF path. You must configure strict SPF SID on each node that supports segment routing.

  The algorithm 1 is identical to algorithm 0, but it requires all the nodes along the path to honor the SPF routing decision. Local policy does not alter the forwarding decision. For example, a packet is not forwarded through locally engineered path.

Why Strict SPF

In the case of link or node failure in the tunnel path, it is possible that MPLS traffic routed via the SR-TE tunnel is rerouted back to the tunnel head end from mid chain if the traffic is diverted to the repair path. If the head-end routes this MPLS traffic via the SR-TE tunnel again, then the same MPLS traffic may loop along the tunnel until TTL expires, even though there is an alternate IGP shortest path to the destination is available.

The strict SPF SID can prevent the looping of traffic through SR-TE tunnels. With strict SPF support, every router is configured to have both default SID, that is, SID0 and strict-SPF SID, that is, SID1. If the tunnel traffic is routed back to the head-end, it arrives at head end with strict-SPF SID as active label, which gets forwarded via the non-tunnel IGP shortest path (native path), thus breaking the looping along the SR-TE tunnel. Strict SPF prefix-SIDs are preferred over default prefix SIDs for SRTE tunnel when all nodes in the area/tunnel-path are Strict SPF capable.

Strict-SPF Capability Advertisement

OSPF advertises the strict SPF capability in SR-Algorithm TLV of the Router Information (RI) opaque link state advertisements (LSA), when segment routing is enabled globally or on a specific area. OSPF includes both algorithm 0 (SPF) and algorithm 1 (Strict-SPF SID) in the SR-Algorithm TLV.

When received, OSPF parses the router information opaque LSA to find the SR Algorithm TLV. If the TLV is missing or algorithm 1 is not included in the TLV, OSPF ignores all strict-SPF SID advertisements from the advertisement router.

OSPF continues to support only single SRGB. The same SRGB is used for both regular SIDs and strict-SPF SIDs. Like regular SID, OSPF must not use out of SRGB range strict-SPF SIDs.

Strict-SPF SID Advertisement in Extended Prefix LSA

OSPF advertises the strict SPF SID connected maps in prefix SID sub-TLV with algorithm set to 1 in OSPF extended prefix TLV of extended prefix opaque LSA. Both default SID and strict SPF SID for the same prefix are advertised in the same LSA. OSPF advertise separate explicit-NULL for regular and strict-SPF SIDs. Both the SIDs share same attach flag.

OSPF advertises the strict SPF SID mapping server entries in Prefix SID Sub-TLV with algorithm set to 1, in OSPF Extended Prefix Range TLV of Extended prefix opaque LSA. Both default SID and strict SPF SID may be advertised for the same prefix. If multiple SIDs of same algorithm are advertised for the same prefix, the receiving router uses the first encoded SID. OSPF advertise separate explicit-NULL for regular and strict-SPF SIDs. Both SIDs share same attach flag. The setting of attach flag in the regular SID takes over precedence if they differ.

If the SR-Algorithm TLV is missing or algorithm 1 is not included in the TLV, OSPF ignores all strict-SPF SID advertisements from the advertisement router. If multiple SIDs of same algorithm are received for the same prefix, the receiving router uses the first encoded SID.If the Explicit-NULL and Attach flags differ for the received SID0 & SID1 of a prefix, then the flags of SID0 takes over precedence.

Interaction with SR-TE and Router Information Base

Like default SID, the strict-SPF SID also communicates with SR-TE only if SR and TE both are enabled for that area. There are three forms of communications that might happen with SR-TE related to strict-SPF SID:

- OSPF announces to SR-TE whether the area is strict-SPF capable or incapable. An area is strict SPF capable, if all the nodes in the area are strict spf capable and each node has at least one strict SPF SID configured.
- OSPF announces to SR-TE the strict-SPF SIDs for all prefixes and registered prefix paths.
- SR-TE prefers strict SPF SID for the label stack. OSPF receives tunnel list from SR-TE whenever there is a change to the list of auto-route announce tunnel list. For each tunnel, SR-TE indicates whether the tunnel is created using strict-SPF SIDs or default SIDs. OSPF runs full SPF whenever the updated tunnel list is received from SR-TE and replaces the RIB paths of prefixes reachable via the tunnel endpoint to the tunnel next hop.

Strict-SPF SIDs are not installed in the router information base (RIB). Only default SIDs get installed as the outgoing labels for the prefixes installed in the RIB. Both SR-TE tunnel types are installed in the RIB.

Enabling and Disabling OSPFv2 Segment Routing Strict SPF

The strict SPF feature is enabled by default when segment-routing mpls is configured under OSPF and global mode. There is no separate CLI to enable or disable it.

Configuring OSPFv2 Segment Routing Strict SPF SID

Perform the following steps to configure OSPFv2 segment routing strict SPF.

```
segment-routing mpls
 connected-prefix-sid-map
  address-family ipv4
   10.0.0.0/8 2
   172.16.0.0/8 3
  address-family ipv4 strict-spf
   10.0.0.0/8 22
   172.16.0.0/8 23
  exit-address-family
```

Verifying OSPFv2 Segment Routing Strict SPF

Use the following commands to verify OSPFv2 segment routing strict SPF.

Verifying OSPFv2 Segment Routing Strict SPF SID

```
Device#show ip ospf database opaque-area type ext-prefix

            OSPF Router with ID (10.0.0.4) (Process ID 10)

                Type-10 Opaque Area Link States (Area 0)

  LS age: 40
  Options: (No TOS-capability, DC)
  LS Type: Opaque Area Link
  Link State ID: 10.7.0.3
  Opaque Type: 7 (Extended Prefix)
  Opaque ID: 3
  Advertising Router: 10.0.0.2
  LS Seq Number: 80000003
  Checksum: 0xFB42
  Length: 56

    TLV Type: Extended Prefix
    Length: 32
      Prefix    : 10.0.0.6/32
      AF        : 0
      Route-type: Intra
      Flags     : N-bit

      Sub-TLV Type: Prefix SID
      Length: 8
        Flags : None
        MTID  : 0
        Algo  : SPF
        SID   : 100

      Sub-TLV Type: Prefix SID
      Length: 8
        Flags : None
        MTID  : 0
```

```
        Algo  : Strict SPF
        SID   : 101

Device#show ip ospf segment-routing sid-database


            OSPF Router with ID (10.0.0.4) (Process ID 10)

OSPF Segment Routing SIDs

Codes: L - local, N - label not programmed,
       M - mapping-server

SID             Prefix              Adv-Rtr-Id       Area-Id  Type      Algo
--------------  ------------------  ---------------  -------  --------  ----
2               10.0.0.2/32          10.0.0.2          0        Intra     0
4       (L)     10.0.0.4/32          10.0.0.4          0        Intra     0
7               10.0.0.7/32          10.0.0.5          0        Intra     0
9               10.0.0.8/32          10.0.0.2          0        Intra     0
20              10.0.2.20/32         10.2.2.2          0        Intra     0
21              10.0.22.21/32        10.2.2.2          0        Intra     1
22      (M)     10.0.2.22/32                                    Unknown   0
29      (M)     10.0.22.29/32                                   Unknown   1
33              10.0.33.33/32        10.3.3.3          0        Intra     1
38      (M)     10.0.3.38/32                                    Unknown   0
39      (M)     10.0.33.39/32                                   Unknown   1
77              10.77.77.77/32       10.5.5.5          0        Inter     0
92      (M)     10.1.2.92/32                                    Unknown   0
99              10.99.99.99/32       10.9.9.9          0        Intra     0
100             10.0.2.100/32        10.2.2.2          0        Intra     0
101             10.0.2.100/32        10.2.2.2          0        Intra     1
120             10.3.3.120/32        10.3.3.3          0        Intra     0
121             10.3.3.120/32        10.3.3.3          0        Intra     1

Device#show ip ospf segment-routing mapping-server

            OSPF Router with ID (10.0.0.4) (Process ID 10)

Advertise local: Enabled
Receive remote: Enabled

Flags: i - sent to mapping-server, u - unreachable,
       s - self-originated

10.0.2.22/32 (R), range size 1
   Adv-rtr        Area          LSID          SID    Type    Algo
i  10.2.2.2        0            10.0.0.4       22     Intra   0
s  10.4.4.4        24           10.0.0.1       22     Inter   0

10.1.2.92/32 (R), range size 1
   Adv-rtr        Area          LSID          SID    Type    Algo
i  10.2.2.2        0            10.0.0.5       92     Intra   0
s  10.4.4.4        24           10.0.0.2       92     Inter   0

10.0.3.38/32 (R), range size 1
   Adv-rtr        Area          LSID          SID    Type    Algo
i  10.3.3.3        0            10.0.0.2       38     Intra   0
s  10.4.4.4        24           10.0.0.3       38     Inter   0

10.3.3.48/32 (R), range size 1
   Adv-rtr        Area          LSID          SID    Type    Algo
i  10.3.3.3        0            10.0.0.3       48     Intra   0
s  10.4.4.4        24           10.0.0.4       48     Inter   0
```

```
10.0.22.29/32 (R), range size 1
   Adv-rtr        Area          LSID          SID    Type    Algo
i  10.2.2.2        0            10.0.0.6       29     Intra   1
s  10.4.4.4        24           10.0.0.5       29     Inter   1

10.1.22.99/32 (R), range size 1
   Adv-rtr        Area          LSID          SID    Type    Algo
i  10.2.2.2        0            10.0.0.7       99     Intra   1
s  10.4.4.4        24           10.0.0.6       99     Inter   1

10.0.33.39/32 (R), range size 1
   Adv-rtr        Area          LSID          SID    Type    Algo
i  10.3.3.3        0            10.0.0.4       39     Intra   1
s  10.4.4.4        24           10.0.0.7       39     Inter   1

10.3.33.49/32 (R), range size 1
   Adv-rtr        Area          LSID          SID    Type    Algo
i  10.3.3.3        0            10.0.0.5       49     Intra   1
s  10.4.4.4        24           10.0.0.8       49     Inter   1
```

```
Device#show ip ospf segment-routing local-prefix
            OSPF Router with ID (10.0.0.7) (Process ID 10)
Area 0:
 Prefix:             Sid:   Index:          Type:      Algo: Source:
 10.2.2.2/32          2      10.0.0.0         Intra      0     Loopback0
                      22     10.0.0.0         Intra      1     Loopback0
 10.23.23.4/32        233    10.0.0.1         Intra      1     Loopback3
```

Verifying OSPFv2 Segment Routing Strict SPF Capability

```
Device#show ip ospf database opaque-area type router-information self

            OSPF Router with ID (10.0.0.4) (Process ID 10)

                Type-10 Opaque Area Link States (Area 0)

  LS age: 1692
  Options: (No TOS-capability, DC)
  LS Type: Opaque Area Link
  Link State ID: 10.4.0.0
  Opaque Type: 4 (Router Information)
  Opaque ID: 0
  Advertising Router: 10.4.4.4
  LS Seq Number: 80000002
  Checksum: 0x72B
  Length: 60

    TLV Type: Router Information
    Length: 4
    Capabilities:
      Graceful Restart Helper
      Stub Router Support
      Traffic Engineering Support

    TLV Type: Segment Routing Algorithm
    Length: 2
      Algorithm: SPF
      Algorithm: Strict SPF

    TLV Type: Segment Routing Range
    Length: 12
```

```
      Range Size: 8000

      Sub-TLV Type: SID/Label
      Length: 3
        Label: 16000

    TLV Type: Segment Routing Node MSD
    Length: 2
      Sub-type: Node Max Sid Depth, Value: 10
```

Verifying Strict SPF Labels Used in OSPF Local RIB Database

```
Device#show ip ospf rib 10.0.0.8

            OSPF Router with ID (10.0.0.6) (Process ID 10)


                Base Topology (MTID 0)

OSPF local RIB
Codes: * - Best, > - Installed in global RIB
LSA: type/LSID/originator

*>  10.0.2.100/32, Intra, cost 21, area 0
     SPF Instance 28, age 00:01:19
      contributing LSA: 10/10.7.0.3/10.2.2.2 (area 0)
     SID: 100, Properties: Sid, LblRegd, SidIndex, N-Flag, TeAnn
     Strict SPF SID: 101, Properties: Force, Sid, LblRegd, SidIndex, N-Flag
     Flags: RIB, HiPrio
      via 10.6.0.3, Ethernet0/1, label 16100, strict label 16101
       Flags: RIB
       LSA: 1/10.2.2.2/10.2.2.2
      PostConvrg repair path via 10.6.0.5, Ethernet0/3, label 16100, strict label 16100,
cost 31
       Flags: RIB, Repair, PostConvrg, IntfDj, BcastDj
       LSA: 1/10.2.2.2/10.2.2.2
```

Verifying Strict SPF TILFA Tunnels

```
Device#show ip ospf fast-reroute ti-lfa tunnels internal

            OSPF Router with ID (10.0.0.2) (Process ID 10)

                        Area with ID (0)

                    Base Topology (MTID 0)


TI-LFA Release Node Tree:

TI-LFA Release Node 10.4.4.4 via 10.2.0.1 Ethernet0/0, instance 12, metric 20
  Interface MPLS-SR-Tunnel2
    Tunnel type: MPLS-SR (strict spf)
    Tailend router ID: 10.4.4.4
    Termination IP address: 10.4.4.4
    Outgoing interface: Ethernet0/0
    First hop gateway: 10.2.0.1
    instance 12, refcount 1
      rn-1: rtrid 10.4.4.4, addr 10.4.4.4, strict node-sid label 16044

TI-LFA Release Node 10.4.4.4 via 10.3.0.3 Ethernet0/1, instance 12, metric 20
  Interface MPLS-SR-Tunnel1
```

```
    Tunnel type: MPLS-SR (strict spf)
    Tailend router ID: 10.4.4.4
    Termination IP address: 10.4.4.4
    Outgoing interface: Ethernet0/1
    First hop gateway: 10.3.0.3
    instance 12, refcount 1
      rn-1: rtrid 10.4.4.4, addr 10.4.4.4, strict node-sid label 16044


TI-LFA Node Tree:

TI-LFA Node 10.1.1.1 via 10.2.0.1 Ethernet0/0, abr, instance 12, rspt dist 0
  not-in-ext-p-space, in-q-space, interesting node 1
  Link Protect strict Path-1: via 10.3.0.3 Et0/1, parent 1/10.4.4.4, metric:30,
rls-pt:10.4.4.4 at dist:20
  repair:y, rn-cnt:1, first-q:10.4.4.4, rtp-flags:Repair, PostConvrg, IntfDj
    rn-1: rtrid 10.4.4.4, addr 10.4.4.4, strict node-sid label 16044
    Protected by: MPLS-SR-Tunnel1, tailend 10.4.4.4, rls node 10.4.4.4
    instance 12, metric 20, refcount 1


TI-LFA Node 10.3.3.3 via 10.3.0.3 Ethernet0/1, instance 12, rspt dist 0
  not-in-ext-p-space, in-q-space, interesting node 1
  Link Protect strict Path-1: via 10.2.0.1 Et0/0, parent 1/10.4.4.4, metric:30,
rls-pt:10.4.4.4 at dist:20
  repair:y, rn-cnt:1, first-q:10.4.4.4, rtp-flags:Repair, PostConvrg, IntfDj
    rn-1: rtrid 10.4.4.4, addr 10.4.4.4, strict node-sid label 16044
    Protected by: MPLS-SR-Tunnel2, tailend 10.4.4.4, rls node 10.4.4.4
    instance 12, metric 20, refcount 1

TI-LFA Node 10.4.4.4 via 10.2.0.1 Ethernet0/0, abr, instance 12, rspt dist 10
  in-ext-p-space, in-q-space, interesting node 1
  Link Protect strict Path-1: via 10.3.0.3 Et0/1, parent 1/10.3.3.3, metric:20,
rls-pt:10.3.3.3 at dist:10
  repair:y, rn-cnt:0, first-q:10.4.4.4, rtp-flags:Repair, PostConvrg, IntfDj, PrimPath
    Protected by: directly connected TI-LFA

TI-LFA Node 10.4.4.4 via 10.3.0.3 Ethernet0/1, abr, instance 12, rspt dist 10
  in-ext-p-space, in-q-space, interesting node 1
  Link Protect strict Path-1: via 10.2.0.1 Et0/0, parent 1/10.1.1.1, metric:20,
rls-pt:10.1.1.1 at dist:10
  repair:y, rn-cnt:0, first-q:10.4.4.4, rtp-flags:Repair, PostConvrg, IntfDj, PrimPath
    Protected by: directly connected TI-LFA


TI-LFA Protected neighbors:

  Neighbor 10.2.0.1 Ethernet0/0, ID 10.1.1.1, Dist 10, instance 12
    TI-LFA Required, TI-LFA Computed, RLFA not Required
    TI-LFA protection Required: link

  Neighbor 10.3.0.3 Ethernet0/1, ID 10.3.3.3, Dist 10, instance 12
    TI-LFA Required, TI-LFA Computed, RLFA not Required
    TI-LFA protection Required: link
```

Verifying Strict SPF SR-TE Tunnels

```
Device#show mpls traffic-eng segment-routing ospf summary
IGP Area[1]:  ospf 10  area 0, Strict SPF Enabled:
Nodes:
IGP Id: 10.1.1.20, MPLS TE Id: 10.1.1.1, OSPF area 0
```

```
   2 links with segment-routing adjacency SID
IGP Id: 10.2.0.0, MPLS TE Id: 10.2.2.2, OSPF area 0
   2 links with segment-routing adjacency SID
IGP Id: 10.3.0.0, MPLS TE Id: 10.3.3.3, OSPF area 0
   3 links with segment-routing adjacency SID
IGP Id: 10.4.4.4, MPLS TE Id: 10.4.4.4, OSPF area 0
   3 links with segment-routing adjacency SID
IGP Id: 10.5.0.0, MPLS TE Id: 10.5.5.5, OSPF area 0
   2 links with segment-routing adjacency SID
Prefixes:
10.1.1.1/32, SID index: 1, Strict SID index: 11
10.2.0.2/32
10.2.2.2/32, SID index: 2, Strict SID index: 22
10.2.2.22/32, SID index: 222, Strict SID index: 2222
10.3.3.3/32, SID index: 3, Strict SID index: 34
10.3.3.33/32, SID index: 333, Strict SID index: 1333
10.4.4.4/32, SID index: 4, Strict SID index: 444
10.5.5.5/32, SID index: 5, Strict SID index: 555
10.6.6.6/32, SID index: 6
10.7.7.7/32, SID index: 7
Total:
  Node Count         : 5
  Adjacency-SID Count: 17
  Prefix-SID Count   : 10
Grand Total:
  Node Count         : 5
  Adjacency-SID Count: 17
  Prefix-SID Count   : 10
  IGP Areas Count    : 1
```

Verifying Protected adj-SIDs Using Strict SPF Repair Path

```
Device#sh ip ospf segment-routing protected-adjacencies detail

            OSPF Router with ID (10.0.0.0) (Process ID 10)

                        Area with ID (0)

Nbr id 10.0.0.1, via 10.0.0.2 on Ethernet0/1, Label 26
  Primary path: via 10.0.0.2 on Et0/1, out-label 3
  Repair path: via 10.0.0.3 on Et0/2, out-label 13222, cost 31, labels 0
  Nbr Prefix 10.0.0.4, Strict
Nbr id 10.0.0.5, via 10.0.0.3 on Ethernet0/2, Label 25
  Primary path: via 10.0.0.3 on Et0/2, out-label 3
  Repair path: via 10.0.0.2 on Et0/1, out-label 12333, cost 21, labels 0
  Nbr Prefix 10.0.0.5, Strict
```

Verifying Segment Routing Global Block

```
Device#show ip ospf segment-routing global-block

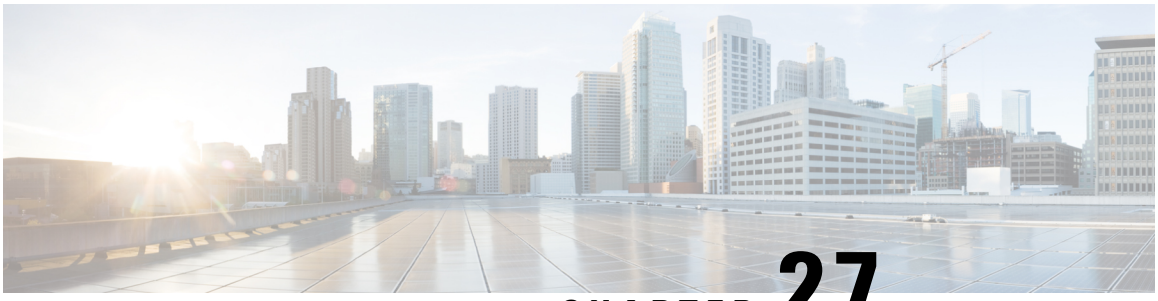
            OSPF Router with ID (10.0.0.0) (Process ID 10)

OSPF Segment Routing Global Blocks in Area 0

  Router ID:      SR Capable: SR Algorithm: SRGB Base: SRGB Range:  SID/Label:

 *10.0.0.0         Yes         SPF,StrictSPF 16000      8000         Label
  10.0.0.1         Yes         SPF,StrictSPF 16000      8000         Label
  10.0.0.2         Yes         SPF,StrictSPF 16000      8000         Label
  10.0.0.3         Yes         SPF           16000      8000         Label
  10.0.0.4         Yes         SPF,StrictSPF 16000      8000         Label
```

```
  10.0.0.5         No
  10.0.0.6         Yes         SPF           16000      8000         Label
Device#
```



CHAPTER 27

Segment Routing OSPFv2 Microloop Avoidance

The feature enables link-state routing protocols such as IS-IS and OSPF to prevent or avoid microloops during network convergence after a topology undergoes any change.

- Feature Information for Segment Routing OSPFv2 Microloop Avoidance, on page 233
- Information About Segment Routing OSPFv2 Microloop Avoidance, on page 234
- Prerequisites for Segment Routing OSPFv2 Microloop Avoidance, on page 237
- Restrictions for Segment Routing OSPFv2 Microloop Avoidance, on page 238
- Configuring Segment Routing OSPFv2 Microloop Avoidance, on page 238
- Verifying Segment Routing OSPFv2 Microloop Avoidance, on page 238

Feature Information for Segment Routing OSPFv2 Microloop Avoidance

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 26: Feature Information for Segment Routing OSPFv2 Microloop Avoidance

Feature Name	Releases	Feature Information
Segment Routing OSPFv2 Microloop Avoidance	Cisco IOS XE Amsterdam 17.3.2	The Segment Routing microloop avoidance feature enables link-state routing protocols such as IS-IS and OSPF to prevent or avoid microloops during network convergence after a topology change. The following commands was introduced/modified by this feature: **microloop avoidance segment-routing**.

Information About Segment Routing OSPFv2 Microloop Avoidance

Microloops are brief packet loops that occur in the network following a topology change (link down, link up, or metric change events). Microloops are caused by the non-simultaneous convergence of different nodes in the network. If nodes converge and send traffic to a neighbor node that has not converged, traffic may be looped between these two nodes, resulting in packet loss, jitter, and out-of-order packets.

If segment routing microloop avoidance feature detects a topology change, it creates a loop-free path to the destination using a list of segments.

Microloops

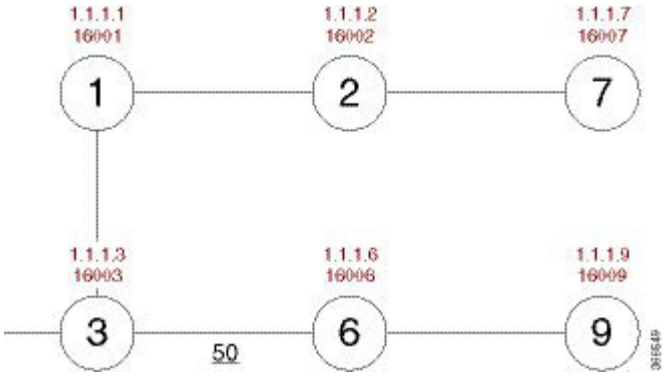
When changes occur in a network topology because of the failure or restoration of a link or a network device, IP Fast Reroute enables rapid network convergence by moving traffic to precomputed backup paths until regular convergence mechanisms move traffic to a newly computed best path, which is also known as a post-convergence path. This network convergence may cause short microloops between two directly or indirectly connected devices in the topology. Microloops are caused when different nodes in the network calculate alternate paths at different times and independently of each other. For instance, if a node converges and sends traffic to a neighbor node, which has not converged yet, traffic may loop between the two nodes.

Microloops may or may not result in traffic loss. If the duration of a microloop is short, that is the network converges quickly, packets may loop for a short duration before their time-to-live (TTL) expires. Eventually, the packets will get forwarded to the destination. If the duration of the microloop is long, that is one of the routers in the network is slow to converge, packets may expire their TTL or the packet rate may exceed the bandwidth, or the packets might be out of order, and packets may dropped.

Microloops that are formed between a failed device and its neighbors are called local uloops, whereas microloops that are formed between devices that are multiple hops away are called remote uloops. Local uloops are usually seen in networks where local loop-free alternate (LFA) path is not available. In such networks, remote LFAs provide backup paths for the network.

The information discussed above can be illustrated with the help of an example topology.

Figure 28: Microloop Example Topology



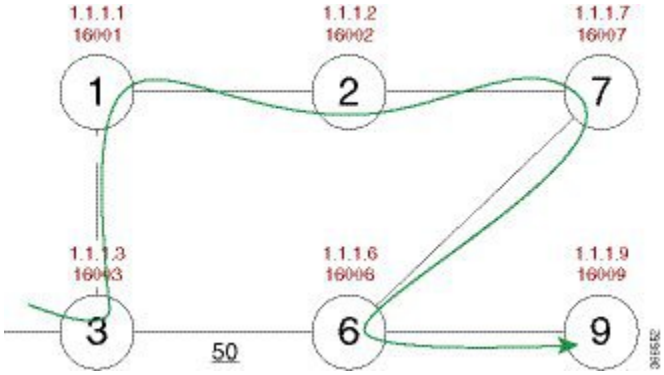
The assumptions in this example are as follows:

- The default metrics is 10 for each link except for the link between Node 3 and Node 6, which has a metric of 50. The order of convergence with SPF backoff delays on each node is as follows:
  - Node 3—50 milliseconds
  - Node 1—500 milliseconds
  - Node 2—1 second
  - Node 7—1.5 seconds

A packet sent from Node 3 to Node 9, the destination, traverses via Node 6.

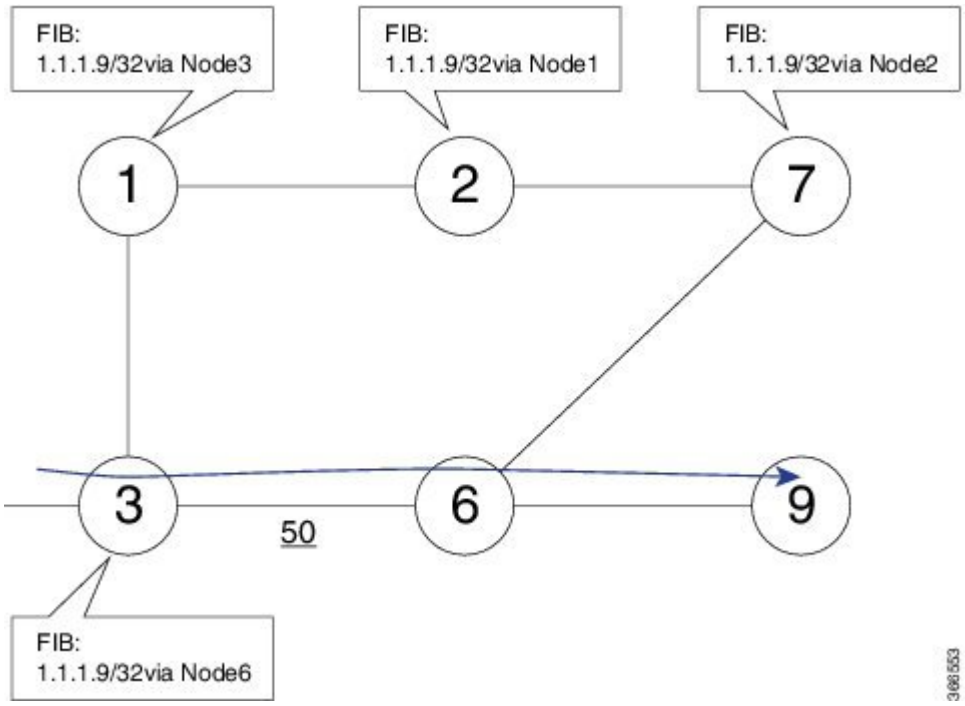
If a link is established between Node 6 and Node 7, the shortest path for a packet from Node 3 to Node 9 would be Node 1, Node 2, Node 7, and Node 6 before the packet reaches the destination, Node 9.

Figure 29: Microloop Example Topology—Shortest Path



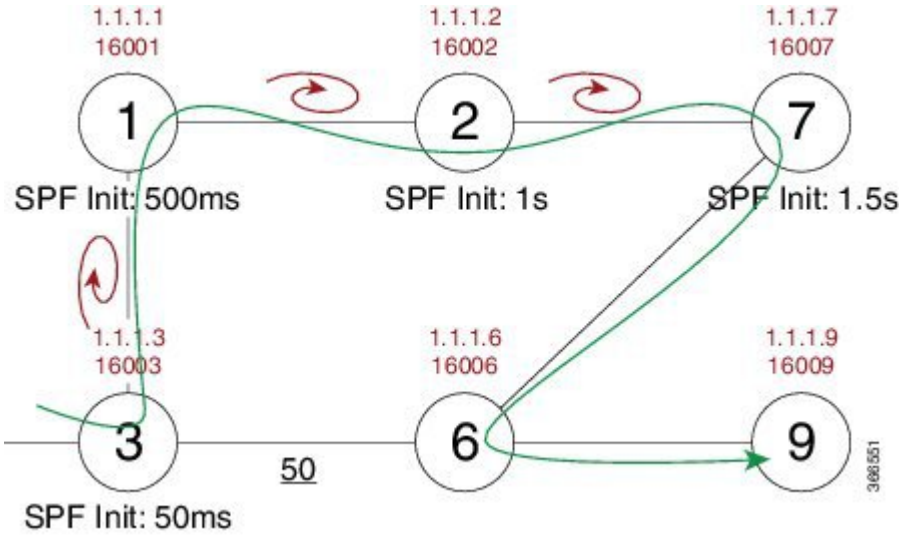
The following figure shows the Forwarding Information Base (FIB) table in each node before the link between Node 6 and Node 7 is established. The FIB entry contains the prefix of the destination node (Node 9) and the next hop.

Figure 30: Microloop Example Topology—FIB Entry



When the link between Node 6 and Node 7 comes up, microloops occur for the links based on the order of convergence of each node. In this example, Node 3 converges first with Node 1 resulting in a microloop between Node 3 and Node 1. Then, Node 1 converges next resulting in a microloop between Node 1 and Node 2. Next, Node 2 converges next resulting in a microloop between Node 2 and Node 7. Finally, Node 7 converges resolving the microloop and the packet reaches the destination Node 9, as shown in the following figure.

Figure 31: Microloop Example Topology—Microloops

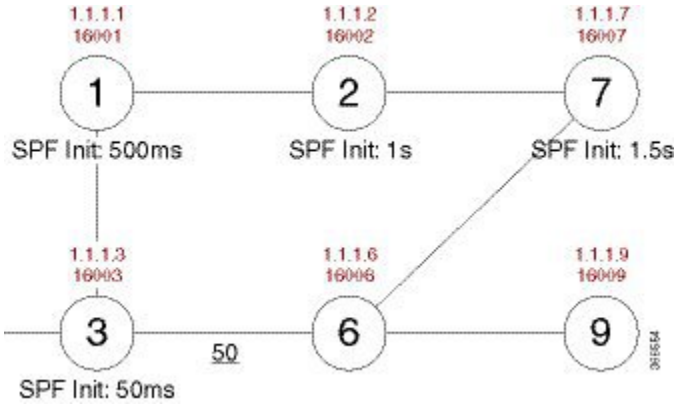


Adding the SPF convergence delay, microloop results in a loss of connectivity for 1.5 seconds, which is the convergence duration specified for node 7.

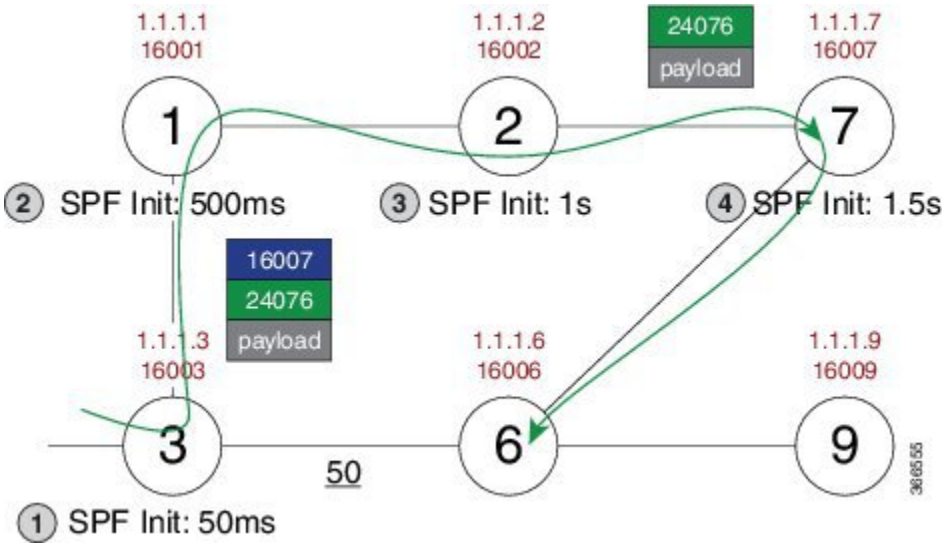
Preventing Microloops using Segment Routing

This section explains how segment routing prevents microloops using an example. Node 3 in the example is enabled with the **microloop avoidance segment-routing** command.

Figure 32: Microloop Example Topology—Segment Routing



Instead of updating the FIB table, Node 3 builds a dynamic loop-free path to the destination (Node 9) using a list of segments IDs, which include the prefix segment ID (SID) of Node 7, which is 16007, and the adjacency segment ID (SID) of Node 6, which is 24076.



So the packet from Node 3 reaches its destination Node 9 without the risk of microloop until the network converges. Finally, Node 3 updates the FIB with the new path.

Prerequisites for Segment Routing OSPFv2 Microloop Avoidance

Before configuring SR microloop avoidance, ensure that the segment routing is globally configured in the OSPF router mode.

```
router ospf process
segment-routing mpls
```

Restrictions for Segment Routing OSPFv2 Microloop Avoidance

- Segment Routing OSPFv2 microloop avoidance does not support Multi Topology Routing (MTR). It supports only MTID 0.
- A list of segment IDs along the post convergence path is used only if the nodes in the the list are SR capable and have atleast one node SID. Otherwise, OSPF installs the post convergence path immediately.
- SR microloop avoidance is used for link up, link down, and link metric change events of point-to-point interfaces and broadcast interfaces with two neighbors only.
- SR microloop avoidance can be used only for one topology change. When multiple topology changes occur, OSPF installs the post convergence path immediately.

Configuring Segment Routing OSPFv2 Microloop Avoidance

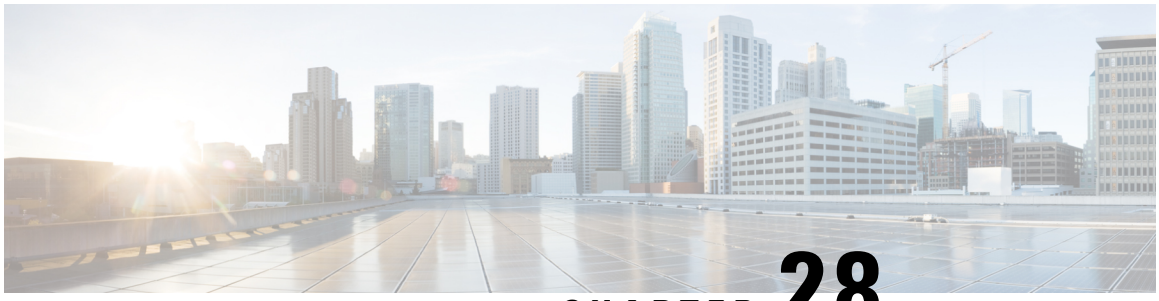
Enables segment routing microloop avoidance for all the prefixes.

```
router ospf
 microloop avoidance segment-routing
 microloop avoidance rib-update-delay delay-time
```

The **microloop avoidance rib-update-delay** *delay-time* command is used to configure the delay in milliseconds for a node to wait before updating the node's forwarding table and stops using the microloop avoidance. The default value for the RIB delay is 5000 milliseconds.

Verifying Segment Routing OSPFv2 Microloop Avoidance

Use the **show ip ospf segment-routing microloop avoidance** command to check if SR microloop avoidance is enabled or not.



CHAPTER 28

Performance Measurement for Traffic Engineering

Metrics such as packet loss, delay, delay variation (jitter) and bandwidth utilization help you evaluate the performance of your network. You can use these metrics as input for Traffic Engineering (TE) and direct the flow of traffic through the network to conform to Service Level Agreements (SLAs). With this feature, you can configure the measurement and advertisement of link delay metrics for TE.

- Feature Information for Performance Measurement for Traffic Engineering, on page 239
- Information about Performance Metrics for Traffic Engineering, on page 240
- How to Configure Performance Measurement for Traffic Engineering, on page 244
- Additional References, on page 249

Feature Information for Performance Measurement for Traffic Engineering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 27: Feature Information for Performance Measurement for Traffic Engineering

Feature Name	Releases	Feature Information
Link Delay Measurement	Cisco IOS XE Bengaluru 17.4	Metrics such as packet loss, delay, delay variation (jitter) and bandwidth utilization help you evaluate the performance of your network. You can use these metrics as input for Traffic Engineering (TE) and direct the flow of traffic through the network to conform to Service Level Agreements (SLAs). With this feature, you can configure the measurement and advertisement of link delay metrics for TE.

Information about Performance Metrics for Traffic Engineering

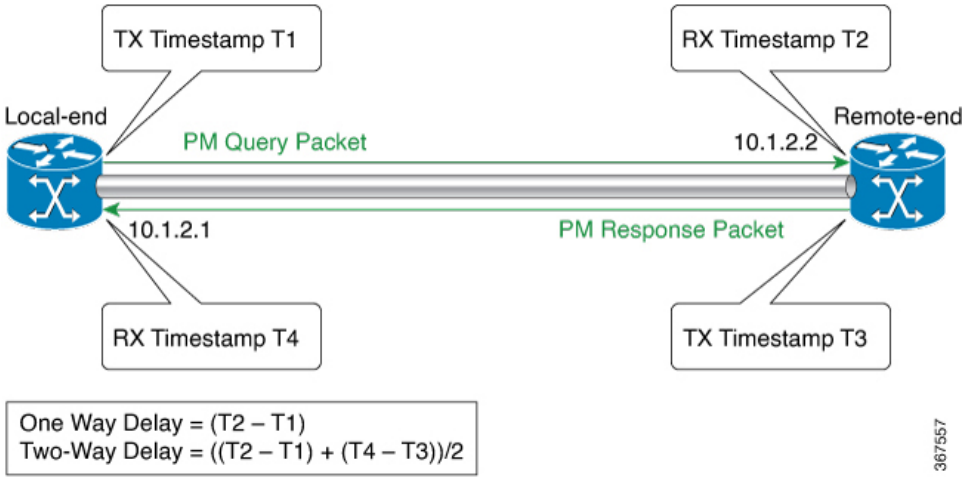
Overview of Link Delay Measurement

Link delay is measured using PM Query packets in the format defined in RFC 6374. To support the packet format, the remote line card must be MPLS capable.

Note Only two-way link delay measurement is supported.

For link delay measurement, an MPLS multicast MAC address is used to send delay measurement probe packets to next-hops. You need not configure next-hop addresses for the links. The remote side line card must support the MPLS multicast MAC address.

The following figure shows the measurement of link delay using the PM Query and Response packets.



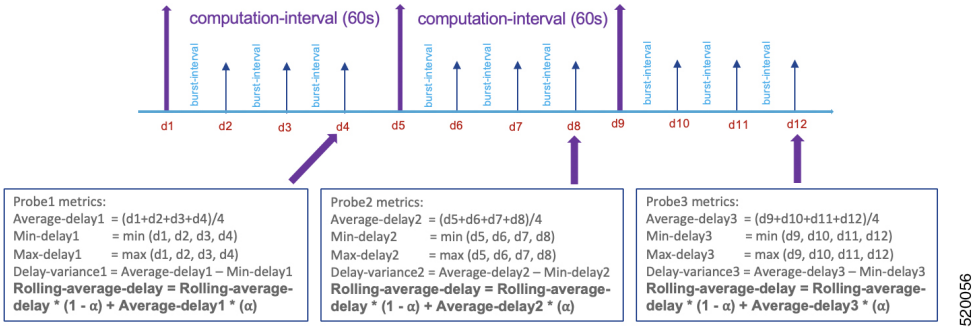
1. The local-end router sends a burst of PM Query packets to the remote-end router at the configured interval. The packets are timestamped (T1) before they are sent.
2. At the remote-end router, packets are timestamped (T2) when they are received.
3. The remote-end router sends the PM packets containing the timestamps (T1 and T2) back to the local-end router. The packets are timestamped (T3) before they are sent.
4. At the local-end router, packets are timestamped (T4) when they are received.
5. At the local-end router, two-way link delay is measured using the timestamps of the PM packets.

Link Delay Metrics for a Computation Interval

The local-end router sends a configured count of PM query packets to the remote-end router at configured burst intervals. The local-end router measures two-way link delay for each burst of PM Query packets that it sends to the remote-end router and receives back with timestamps.

During each configured probe or computation interval, multiple bursts of PM packets are sent and link delay is measured. Minimum, maximum, and average link delay, and delay variance are calculated for the interval. These metrics are calculated using the link delay that is measured for the bursts sent during the interval.

The following figure illustrates the calculation of delay metrics for a computation interval. In this example, the computation interval is 60 seconds and the burst interval is 15 seconds.



Link Delay Metrics for Advertisement

You can configure the computation and advertisement of delay metrics in a periodic manner, an accelerated manner, or both. The advertisement of link delay metrics is supported with the ISIS, OSPF, and BGP-LS protocols. No additional configuration is required to flood link delay metrics through ISIS, OSPF, and BGP-LS protocols.

Periodic Advertisement

Periodic advertisement is enabled by default. A periodic advertisement interval consists of one or more computation or probe intervals. Link delay metrics are computed at the end of each computation interval. In a periodic advertisement interval, after the last computation interval, the minimum delay computed for a link is compared with the value advertised previously. If the variation in values is beyond configured limits, all the delay metrics for the link are advertised. If the variation in values is within configured limits, the delay metrics for the link are not advertised.

- Suppose a periodic advertisement interval consists of N computation intervals, at the end of a computation interval *i*, the following metrics are computed:
  - Rolling average delay

    Rolling average delay = rolling-average-delay(i-1) * 0.5 + average-delay(i) * 0.5
  - Minimum delay

    Minimum delay = min[min-delay(1),..., min-delay(i-1), min-delay(i)]
  - Maximum delay

    Maximum delay = max[max-delay(1),…, max-delay(i-1), max-delay(i)]
  - Delay variance

    Delay variance = average[delay-variance(1),…, delay-variance(i-1), delay-variance(i)]
- After the last computation interval in the periodic advertisement interval, the minimum delay for a link is compared with the value advertised after the previous interval.

- Case 1: change between the two values is beyond the configured threshold and minimum-change. In this case, all the delay metrics computed for the link after the recent periodic advertisement interval are advertised.
- Case 2: change between the two values is within the configured threshold and minimum-change. In this case, the delay metrics are not advertised.

Accelerated Advertisement

By default, accelerated advertisement is disabled. When you enable accelerated advertisement, the minimum link delay that is computed for a link after a computation interval is compared with the value previously advertised. If the variation in values is beyond configured limits, all the delay metrics for the link are advertised. If the variation in values is within configured limits, the delay metrics for the link aren't advertised.

When link delay metrics are advertised in an accelerated manner, the periodic advertisement interval is reset. This reset ensures the configured interval of time between the recent advertisement and the next periodic assessment.

Link Delay Metrics when the Link State Changes

When a link enters the DOWN state, link delay metrics are advertised with the highest value. The minimum, maximum, and average link delay, and delay variance are advertised with a value of 16.7 seconds (0xFFFFFF). With the highest metric values advertised, routing and SR-TE path computation don't use stale metric values when the link enters the UP state.

Global Link Delay Profile

You can configure a global profile for the measurement of link delay metrics. The profile defines parameters that control the computation and advertisement of link delay metrics and replaces the default configuration. Being global, the profile applies to link delay measurement on all interfaces.

You can configure the following parameter as part of the global profile:

Table 28: Global Link Delay Profile Parameters

Aspect	Parameter	Description
probe	interval	The default probe or computation interval is 30 seconds. The range is 30–3600 seconds.
	protocol	Protocol used to send probes. The default and the only supported protocol is pm-mpls: link delay measurement based on RFC 6374 with MPLS encapsulation.
burst	count	The default value is 10 and range is 1–30.
	interval	The default value is 3000 milliseconds and the range is 30–15000 milliseconds.

Aspect	Parameter	Description
periodic advertisement	interval	The default value is 120 seconds and the interval range is 30–3600 seconds.
	threshold	The default value of periodic advertisement threshold is 10 percent.
	minimum-change	The default value is 1000 microseconds and the range is 0–10000 microseconds.
	disabled	Periodic advertisement is enabled by default.
accelerated advertisement	threshold	The default value is 20 percent and the range is 0–100 percent.
	minimum-change	The default value is 1000 microseconds and the range is 1–100000 microseconds.

Benefits of Link Delay Measurement

You can use link delay metrics such as average, minimum, and maximum delay, and delay variance to determine network latency. Using link delay metrics, you can troubleshoot latency issues or apply Traffic Engineering (TE) solutions to meet Service Level Agreements (SLAs). For example, you could

- configure SR Policies that have acceptable delay
- steer traffic through alternative SR Policies when the delay performance of the serving SR Policies deteriorates beyond acceptable limits.

Restrictions for Link Delay Measurement

Restrictions in IOS XE Release 17.1.x

- Measurement of only two-way link delay is supported.
- PM link delay measurement is based on RFC 6374 and the PM packets use MPLS/GAL encapsulation.
- Only minimum-delay value is used for threshold checks.
- You cannot configure the packet size and TOS/DSCP/EXP of link-delay probe protocol packets.
- Link delay values that exceed two seconds are discarded.

How to Configure Performance Measurement for Traffic Engineering

Configuring Global Link Delay Profile

Configure the parameters of the global link delay profile by entering the interface delay profile mode:

```
performance-measurement
    delay-profile
        interfaces    ---> Global default profile for link delay measurement
            probe
                  computation-interval <seconds> (range:30-3600 seconds; default:30 seconds)
                  burst-interval <milliseconds> (range:30-15000 milliseconds; default: 3000 milliseconds)

                protocol
                    pm-mpls                     Link delay measurement using RFC6374 with MPLS encapsulation
                    twamp-light                 Link delay measurement using RFC5357 [Default]
            advertisement
                periodic                        (default: enabled)
                    disabled
                    interval <seconds>             (range:30-3600 seconds; default:120 seconds)
                    threshold <percentage>         (range:0-100%; default:10%)
                    minimum-change <microseconds>  (range:0-100000 microseconds; default: 1000 microseconds)
                accelerated                     (default: disabled)
                    threshold <percentage>         (range:0-100%; default: 20%)
                    minimum-change <microseconds>  (range:0-100000 microseconds; default: 1000 microseconds)
```

Note For scale scenarios where number of PM Link Delay sessions is higher than 500 sessions, it is recommended to increase burst-interval to the maximum value of 15000 milliseconds.

Configuring Link Delay Measurement for an Interface

Enabling Link Delay Measurement for an Interface

Enable delay-measurement for an interface as follows:

```
performance-measurement
    interface <interface-name>
        delay-measurement
```

Disabling Link Delay Measurement for an Interface

Disable delay-measurement for an interface as follows:

```
performance-measurement
    interface <interface-name>
        no delay-measurement
```

Configuring a Link Delay for an Interface

Set a link delay for an interface as follows:

```
performance-measurement
    interface <interface-name>
        delay-measurement
            advertise-delay <microseconds>    (range: 0-16777215 microseconds)
```

When the advertise-delay is set for an interface,

- the minimum, maximum, and average delays for the associated link are set to the advertise-delay value
- the delay variance for the link is set to zero
- the link delay metrics are immediately advertised.

During the computation interval, PM query and response packets are exchanged and link delay metrics are computed. These metrics are stored in the history buffer and can be accessed using the command **show performance-measurement history interfaces** [**name** *interface-name*] [**adv** | **aggr** | **probe**]. However, when advertise-delay is configured, threshold checks are not performed. Therefore, the computed metrics are not advertised.

Remove the set link delay for an interface as follows:

```
performance-measurement
    interface <interface-name>
        delay-measurement
            no advertise-delay <microseconds>    (range: 0-16777215 microseconds)
```

When the set link delay is removed for an interface,

- delay metrics are unpublished by removing TLVs from the IGP,
- at the end of the subsequent advertisement interval, threshold checks are performed. Based on the threshold checks, link delay metrics are advertised if necessary.

Enabling Monitoring Mode

In the Monitoring Mode, the computed delay metrics are stored in the history buffer. However, the metrics are not advertised by an IGP or BGP-LS. You can display the metrics in the history buffer using the **show performance-measurement history interfaces** [**name** *interface-name*] [**adv** | **aggr** | **probe**] command.

To enable Monitoring Mode, disable both periodic and accelerated advertisement of link delay metrics.



Note Accelerated advertisement is disabled by default.

Disable periodic advertisement as follows:

```
performance-measurement
    delay-profile
        interfaces    ---> Global default profile for link delay measurement
            advertisement
                periodic                        (default: enabled)
                    disabled
```

With Monitoring Mode enabled,

- link delay metrics are not published through Interface Manager attributes in the system.
- link delay metrics are not flooded in the network by IGPs or advertised by BGP-LS.

Verifying Link Delay Configuration

Use the **show performance-measurement summary** [**detail**] command to view the link delay configuration.

Example

```
router#show performance-measurement summary
Total interfaces                    : 2
Maximum PPS                         : 100 pkts/sec

Interface Delay-Measurement:
  Total sessions                    : 2
  Profile configuration:
    Measurement Type                : Two-Way
    Probe interval                  : 30 seconds
    Burst interval                  : 3000 mSec
    Burst count                     : 10 packets
    Protocol                        : MPLS RFC6374
    HW Timestamp Supported          : Yes
    Periodic advertisement          : Enabled
      Interval                      : 120 (effective: 120) sec
      Threshold                     : 10%
      Minimum-Change                : 1000 uSec
    Advertisement accelerated       : Disabled
    Threshold crossing check        : Minimum-delay
  Counters:
    Packets:
      Total sent                    : 289588
      Total received                : 289588
    Errors:
      Total sent errors             : 23
      Total received errors         : 21
        .
        .
        .
```

Viewing Link Delay Information for an Interface

Use the **show performance-measurement interfaces** [**name** *interface-name*] [**detail**] command to view information about the link delay measurement for an interface.

Example

```
router#show performance-measurement interfaces name gigabitEthernet 0/0/7 detail
Interface Name: GigabitEthernet0/0/7 (ifh: 0xF)
  Delay-Measurement           : Enabled
  Local IPV4 Address          : 10.100.1.1
  Local IPV6 Address          : ::
  State                       : Up

  Delay Measurement session:
    Session ID                : 1

    Last advertisement:
      Advertised at: 13:53:11  28 2019 (434548 seconds ago)
      Advertised reason: Periodic timer, min delay threshold crossed
      Advertised delays (uSec): avg: 4011, min: 4033, max: 4050, variance: 4

    Next advertisement:
      Check scheduled in 2 more probes (roughly every 120 seconds)
      Aggregated delays (uSec): avg: 4040, min: 4035, max: 4054, variance: 5
      Rolling average (uSec): 4040

    Current Probe:
      Started at 14:35:38  02 2019 (1 second ago)
```

```
Packets Sent: 1, received: 1
Measured delays (uSec): avg: 4035, min: 4035, max: 4035, variance: 0
Probe samples:
        Packet Rx Timestamp Measured Delay
          14:35:38  02 2019 4035081
Next probe scheduled at 14:36:08  02 2019 (in 29 seconds)
Next burst packet will be sent in 2 seconds
```

Additional Commands

show Commands

*Table 29: **show** Commands for the Local-End Router (Querier)*

Command	Description
show performance-measurement summary [**detail**]	Displays the PM link-delay information, including configuration, session data, and counters.
show performance-measurement interfaces [**name** *interface-name*] [**detail**]	Displays the PM link-delay information for an interface.
show performance-measurement history interfaces [**name** *interface-name*] [**adv** | **aggr** | **probe**]	• **probe** – Displays the PM link-delay probe history for an interfaces. • **adv** – Displays the PM link-delay advertisement history for interfaces. Advertised values of link delay metrics are values flooded using ISIS, OSPF, or BGP. • **aggr** – Displays the PM link-delay aggregated history for interfaces.
show performance-measurement counters interfaces [**name** *interface-name*] [**detail**]	Displays the PM link-delay session counters.
show performance-measurement sessions interface [*session-id*] [**detail**]	Displays information about interfaces that received probe queries from the remote side.

*Table 30: **show** Commands for the Remote-End Router (Responder)*

Command	Description
show performance-measurement responder summary	Displays the PM for link-delay summary on the remote-end router (responder).
show performance-measurement responder interfaces [**name** *interface-name*]	Displays the PM link-delay configuration information for interfaces on the remote-end router.
show performance-measurement responder counters interface [**name** *interface-name*]	Displays the PM link-delay session counters on the remote-end router.

clear Commands

Table 31: clear Commands for the Local-End Router (Querier)

Command	Description
clear performance-measurement all	Clear all performance measurement data, including advertised delay metrics. Using this command withdraws any delay metrics flooded using an IGP or BGP.
clear performance-measurement delay interfaces [**name** *interface-name*]	Clear PM delay information for interfaces. **Note** Using this command withdraws the previously advertised delay for the cleared interfaces. Use this command with care.
clear performance-measurement counters interfaces [**name** *interface-name*]	Clear PM interface counters.
clear performance-measurement counters summary	Clear PM summary counters.

Table 32: clear Commands for the Remote-End Router (Responder)

Command	Description
clear performance-measurement responder counters interfaces [**name** *interface-name*]	Clear PM interface counters on the responder.
clear performance-measurement responder counters summary	Clear PM summary counters on the responder.

debug Commands

Table 33: debug Commands for the Local-End Router (Querier)

Command	Description
debug performance-measurement query [**errors** | **entry** | **packet-errors** | **packets** | **queues** | **timers**]	Enable debug messages on the querier.

Table 34: debug Commands for the Remote-End Router (Responder)

Command	Description
debug performance-measurement responder [**errors** | **entry** | **packet-errors** | **packets** | **queues** | **timers**]	Enable debug messages on the querier.

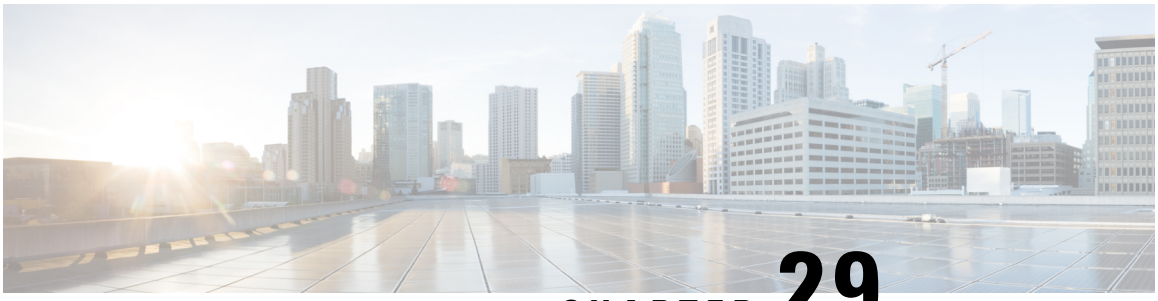
show tech-support Commands

Command	Description
show tech-support perf_measure	Display Performance Measurement related information.
show tech-support monitor event-trace perf_measure	Display trace information related to Performance Measurement.

Additional References

Standards and RFCs

Standard/RFC	Title
RFC 6374	Packet Loss and Delay Measurement for MPLS Networks



CHAPTER 29

Configure Performance Measurement

Table 35: Feature History

Feature Name	Release Information	Description
Segment Routing Performance Measurement Delay Measurement Using RFC 5357 (TWAMP Light)	Cisco IOS XE Bengaluru 17.4	This feature enables hardware timestamping. The Performance Measurement (PM) for link delay uses the light version of Two-Way Active Measurement Protocol (TWAMP) over IP and UDP defined in Appendix I of RFC 5357. TWAMP provides an alternative for interoperability when RFC 6374 is not used.
Segment Routing Absolute One-Way Link Loss Measurement for GRE-IPSec Tunnel	Cisco IOS XE Dublin 17.10.1a	This feature provides a mechanism for link loss measurement for point-to-point GRE-IPSec tunnel, in order to identify paths that meet specified loss criteria.

Network performance metrics such as packet loss, delay, delay variation, and bandwidth utilization are a critical measure for traffic engineering (TE) in service provider networks. These metrics provide network operators with information about 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 the performance measurement (PM) feature to monitor the network metrics for links as well as end-to-end TE label switched paths (LSPs).

The following table explains the functionalities supported by the performance measurement feature for measuring delay for links or SR policies.

Table 36: Performance Measurement Functionalities

Functionality	Details
Probe and burst scheduling	Schedule probes and configure metric advertisement parameters for delay measurement.

Functionality	Details
Metric advertisements	Advertise measured metrics periodically using configured thresholds. Also supports accelerated advertisements using configured thresholds.
Measurement history and counters	Maintain packet delay and loss measurement history and also session counters and packet advertisement counters.

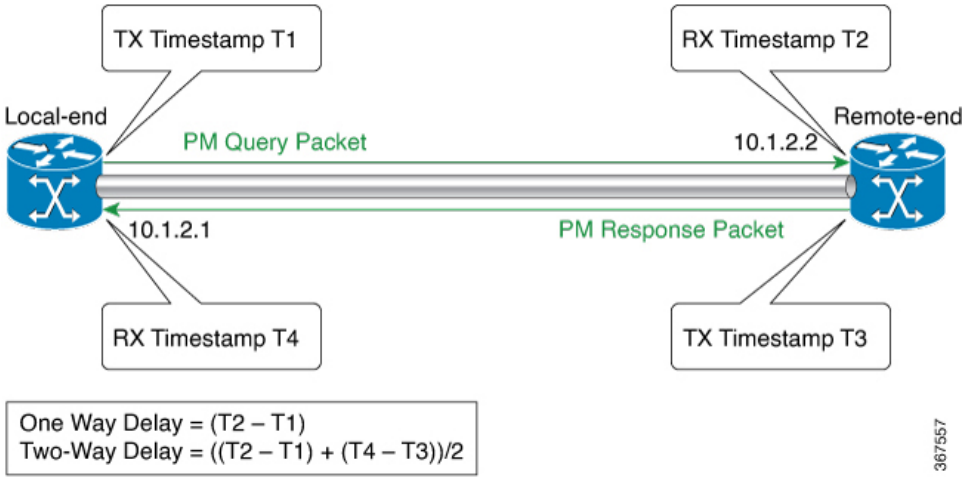
- Link Delay Measurement, on page 252
- End-to-End Delay Measurement, on page 257
- One-Way Link Loss Measurement, on page 260
- Sample show Commands, on page 270

Link Delay Measurement

The PM for link delay uses the light version of Two-Way Active Measurement Protocol (TWAMP) over IP and UDP defined in Appendix I of RFC 5357. Hence, only TWAMP test sessions are implemented and not the TWAMP control protocol. TWAMP provides an alternative for interoperability when RFC 6374 is not used. TWAMP packets are carried over IP and UDP. Thus, the dependency on MPLS dataplane is eliminated.

The following figure explains the PM query and response for link delay.

Figure 33: Performance Measurement for Link Delay



The PM query and response for link delay can be described in the following steps:

1. The local-end router sends PM query packets periodically to the remote side once the egress line card on the router applies timestamps on packets.
2. Ingress line card on the remote-end router applies time-stamps on packets as soon as they are received.
3. The remote-end router sends the PM packets containing time-stamps back to the local-end router. The remote-end router time-stamps the packet just before sending it for two-way measurement.

Restrictions and Usage Guidelines for PM for Link Delay

The following restrictions and guidelines apply for the PM for link delay feature for different links.

- For broadcast links, only point-to-point (P2P) links are supported. P2P configuration on IGP is required for flooding the value.
- As ASR 1000 platforms do not support PTP 1588v2 clock, it can not use the (T2-T1) to calculate the One-way delay value. Therefore, Two-way Delay value is divided by 2 to calculate the One-way delay value.
  - Two-Way Delay = (T2 – T1) + (T4 – T3)
  - One-Way Delay = ((T2 – T1) + (T4 – T3))/2

PM Link Delay: Default Values for Different Parameters

The default values for the different parameters in the PM for link delay is given as follows:

- probe: The default mode for probe is two-way delay measurement.
- interval: The default probe interval is 30 seconds. The range is from 30 to 3600 seconds.
- burst count: The default value is 10 and range is from 1 to 30.
- burst interval: The default value is 3000 milliseconds and the range is from 30 to 15000 milliseconds.
- periodic advertisement: Periodic advertisement is enabled by default.
- periodic-advertisement interval: The default value is 120 seconds and the interval range is from 30 to 3600 seconds.
- periodic-advertisement threshold: The default value of periodic advertisement threshold is 10 percent.
- periodic-advertisement minimum: The default value is 1000 microseconds (usec) and the range is from 0 to 100000 microseconds.
- accelerated advertisement: Accelerated advertisement is disabled by default.
- accelerated-advertisement threshold: The default value is 20 percent and the range is from 0 to 100 percent.
- accelerated-advertisement minimum: The default value is 1000 microseconds and the range is from 1 to 100000 microseconds.

Configuration Example: PM for Link Delay

This example shows how to configure performance-measurement functionalities for link delay as a global default profile.

```
R1(config)#performance-measurement
R1(config-perf-meas)# delay-profile interfaces

R1(config-pm-dm-intf)#advertisement
R1(config-pm-dm-intf-adv)# accelerated // Default: Disabled
```

```
R1(config-pm-dm-intf-adv-acc)#threshold 40 //Default 20%
R1(config-pm-dm-intf-adv-acc)#minimum-change 1000 //Default 1000uSec
R1(config-pm-dm-intf-adv)#periodic
R1(config-pm-dm-intf-adv-per)#interval 100 //Default 120seconds
R1(config-pm-dm-intf-adv-per)#threshold 40 //Default 10%
R1(config-pm-dm-intf-adv-per)#minimum-change 1000 //Default 1000 uSec
R1(config-pm-dm-intf)#probe

R1(config-pm-dm-intf-probe)#computation-interval 40 // Def: 30s
R1(config-pm-dm-intf-probe)#burst-interval 40 // Def: 3000 mSec
R1(config-perf-meas)#delay-profile sr-policy
R1(config-pm-dm-srpol)#advertisement
R1(config-pm-dm-sr-adv)#accelerated // Default: Disabled
R1(config-pm-dm-sr-adv-acc)#threshold 40 //Default 40%
R1(config-pm-dm-sr-adv-acc)#minimum-change 4000 // Def: 500 uSec
R1(config-pm-dm-sr-adv)#periodic

R1(config-pm-dm-srpol-adv-per)#interval 100 // Def: 120 sec
R1(config-pm-dm-srpol-adv-per)#threshold 40 // Def: 10%
R1(config-pm-dm-srpol-adv-per)#minimum-change 2000 // Def: 500 uSec
R1(config-pm-dm-srpol)#probe
R1(config-pm-dm-srpol-probe)#computation-interval 40 // Def: 30s
R1(config-pm-dm-srpol-probe)#burst-interval 40 // Def: 3000 mSec
R1(config-pm-dm-srpol-probe)#exit

R1(config-pm-dm-srpol)#exit

R1(config-pm-dm-srpol-adv-per)#exit
R1 R1(config-pm-dm-intf-probe)#exit
R1(config-pm-dm-intf-adv)#exit

R1(config-pm-dm-intf)#exit

R1(config-perf-meas)#exit
```

This example shows how to enable PM for link delay over an interface.

```
R1(config)#performance-measurement
R1(config-perf-meas)#interface GigabitEthernet 0/0/1
R1(config-pm-intf)#delay-measurement
R1(config-pm-intf-dm)#exit
R1(config-pm-intf-dm)#next-hop ipv4 10.50.62.1
R1(config-pm-intf)#exit
```

Verification: PM Link Delay Configuration

This example shows how to use the **show performance-measurement summary** [**detail**] command to verify the PM for link-delay configuration.

```
R1#show performance-measurement summary detail
Total interfaces                              : 3
Maximum PPS                                   : 100 pkts/sec

Interface Delay-Measurement:
  Total sessions                              : 3
  Profile configuration:
    Measurement Type                          : Two-Way
    Computation interval                      : 30 seconds
    Burst interval                            : 3000 mSec
    Burst count                               : 10 packets
```

```
    Protocol                                : TWAMP-Lite Unauth
    HW Timestamp Supported                  : No
    Periodic advertisement                  : Enabled
      Interval                              : 30 (effective: 30) sec
      Threshold                             : 100%
      Minimum-Change                        : 100000 uSec
    Accelerated advertisement               : Enabled
      Threshold                             : 100%
      Minimum-Change                        : 100000 uSec
    Threshold crossing check                : Minimum-delay
  Counters:
    Packets:
      Total sent                            : 293020
      Total received                        : 293016
    Errors:
        TX:
          Total interface down              : 0
          Total no MPLS caps                : 0
          Total no IP address               : 0
          Total other                       : 19
        RX:
          Total negative delay              : 144
          Total delay threshold exceeded    : 0
          Total missing TX timestamp        : 0
          Total missing RX timestamp        : 0
          Total probe full                  : 0
          Total probe not started           : 0
          Total control code error          : 0
          Total control code notif          : 0
    Probes:
      Total started                         : 29306
      Total completed                       : 29155
      Total incomplete                      : 148
      Total advertisements                  : 3

Global Delay Counters:
  Total packets sent                        : 293020
  Total query packets received              : 293016
  Total invalid session id                  : 0
  Total no session                          : 0

HW Support for MPLS-GAL [RFC6374] timestamp   : No
HW Support for TWAMP [RF5357] timestamp       : No
HW Support for 64 bit timestamp               : No
HW Support for IPv4 UDP Cheksum               : No
```

This example shows how to use the **show performance-measurement interfaces** [*interface-name*] [**detail**] command to verify the PM for link-delay configuration.

```
R1#show performance-measurement interfaces detail
Interface Name: GigabitEthernet0/2/3 (ifh: 0xA)
  Delay-Measurement           : Enabled
  Local IPV4 Address          : 10.50.62.2
  Local IPV6 Address          : ::
  State                       : Up

  Delay Measurement session:
    Session ID                : 1

    Last advertisement:
      Advertised at: 09:21:08  12 2019 (439879 seconds ago)
      Advertised reason: Advertise delay config
      Advertised delays (uSec): avg: 2000, min: 2000, max: 2000, variance: 0
```

```
    Next advertisement:
      Check scheduled at the end of the current probe (roughly every 30 seconds)
      No probes completed
      Rolling average (uSec): 3146

    Current Probe:
      Started at 11:32:17  17 2019 (10 seconds ago)
      Packets Sent: 4, received: 4
      Measured delays (uSec): avg: 1999, min: 1500, max: 2499, variance: 499
      Probe samples:
              Packet Rx Timestamp Measured Delay
                11:32:17  17 2019 1999999
                11:32:20  17 2019 1500000
                11:32:23  17 2019 2499999
                11:32:26  17 2019 1999999
      Next probe scheduled at 11:32:46  17 2019 (in 19 seconds)
      Next burst packet will be sent in 1 seconds
R1#
```

You can also use the following commands for verifying the PM for link delay on the local-end router.

Command	Description
show performance-measurement history interfaces [name*interface-name*] probe	Displays the PM link-delay probe history for interfaces.
show performance-measurement history interfaces [name*interface-name*] aggr	Displays the PM link-delay aggregated history for interfaces.
show performance-measurement counters interface [name*interface-name*] [detail]	Displays the PM link-delay session counters.
show performance-measurement responder interfaces [name*interface-name*]	Displays PM for link-delay for interfaces on the remote-end router.
show performance-measurement responder counters interface [name*interface-name*]	Displays the PM link-delay session counters on the remote-end router.

End-to-End Delay Measurement

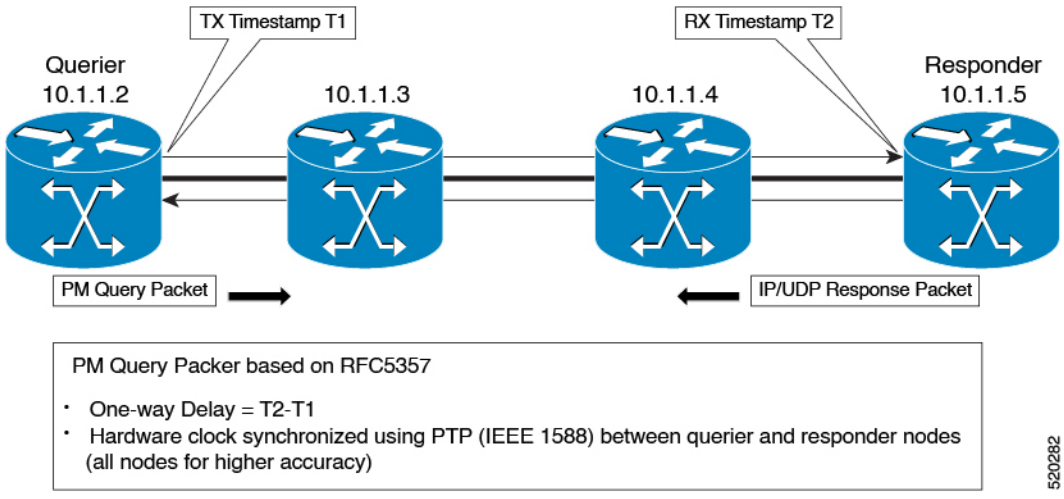
Table 37: Feature History

Feature Name	Release Information	Description
Segment Routing Performance Measurement End-to-End Delay Measurement	Cisco IOS XE Bengaluru 17.4	This feature allows to monitor the end-to-end delay experienced by the traffic sent over a Segment Routing policy. This feature ensures the delay does not exceed the specified threshold value and violate the SLAs. Use this feature to apply extended TE link delay metric (minimum delay value) to compute paths for Segment Routing policies as an optimization metric or as an accumulated delay bound.

Starting from Cisco IOS XE Release 17.4.1, end-to-end delay measurement feature is introduced for Segment Routing Performance Management. Use this feature to monitor the end-to-end delay experienced by the traffic sent over a Segment Routing policy. This feature ensures the delay does not exceed the specified threshold value and violate the SLAs. You can verify the end-to-end delay values before activating the candidate-path or the segment-list of the Segment Routing policy in the forwarding table. You can also use the end-to-end delay values to deactivate the active candidate-path or the segment-list of the Segment Routing Policy in the forwarding table. Use this feature to apply extended TE link delay metric (minimum delay value) to compute paths for Segment Routing policies as an optimization metric or as an accumulated delay bound.

The following figure explains the PM query and response for end-to-end delay measurement.

Figure 34: Performance Measurement for End-to-End Delay Measurement



The PM query and response for end-to-end delay measurement can be described in the following steps:

1. The querier router sends PM query packets periodically to the responder router once the egress line card on the router applies timestamps on packets.
2. Ingress line card on the responder router applies time-stamps on packets when they are received.
3. The end-to-end delay value of an SR Policy is different than the path computation result (the sum of TE link delay metrics) due to several factors like queuing delay within the routers.
4. The remote-end router sends the PM packets containing time-stamps back to the local-end router. The remote-end router time-stamps the packet just before sending it for two-way measurement.
5. The local-end router time-stamps the packet as soon as the packet is received for two-way measurement.

Configuration Example: PM for End-to-End Delay Management

These examples show how to configure on-demand segment routing policy for end-to-end delay management.

```
#show running-config | s on-demand color 800
on-demand color 800 ------------------------------------------------------> SR ODN Policy
authorize
performance-measurement ---------------------------------------------------> SR PM CLI
delay-measurement ---------------------------------------------------------> SR PM CLI
candidate-paths
preference 1
constraints
segments
dataplane mpls
!
!
dynamic
pcep
metric
type delay
!
!
!
#

#show segment-routing traffic-eng policy name *10.216.216.216|800

Name: *10.216.216.216|800 (Color: 800 End-point: 10.216.216.216)
Owners : BGP
Status:
Admin: up, Operational: up for 01:27:24 (since 11-29 04:41:36.053)
Candidate-paths:
Preference 1 (BGP):
Dynamic (pce 10.12.12.12) (active)
Weight: 0, Metric Type: DELAY
Metric Type: DELAY, Path Accumulated Metric: 330
16011 [Prefix-SID, 10.205.205.205]
1133 [Adjacency-SID, 10.50.72.1 - 10.50.72.2]
16009 [Prefix-SID, 10.216.216.216]
Attributes:
Binding SID: 1218
Allocation mode: dynamic
State: Programmed
IPv6 caps enabled
#
```

This example shows how to configure performance-measurement functionalities for end-to-end delay management as a global default profile.

```
R1(config)#performance-measurement
R1(config-perf-meas)#delay-profile sr-policy

R1(config-pm-dm-srpol)#probe
R1(config-pm-dm-srpol-probe)#computation-interval 40

R1(config-pm-dm-srpol-probe)#burst-interval 40

R1(config-pm-dm-srpol-probe)#protocol twamp-light
R1(config-pm-dm-srpol-probe-protocol)#exit

R1(config-pm-dm-srpol-probe)#exit
R1(config-pm-dm-srpol)#advertisement periodic

R1(config-pm-dm-srpol-adv-per)#interval 100
R1(config-pm-dm-srpol-adv-per)#threshold 20

R1(config-pm-dm-srpol-adv-per)#minimum-change 500

R1(config-pm-dm-srpol-adv-per)#exit

R1(config-pm-dm-sr-adv)#exit

R1(config-pm-dm-srpol)#advertisement accelerated

R1(config-pm-dm-sr-adv-acc)#threshold 40

R1(config-pm-dm-sr-adv-acc)#minimum-change 1000

R1(config-pm-dm-sr-adv-acc)#exit
R1(config-pm-dm-sr-adv)#exit

R1(config-pm-dm-srpol)#exit

R1(config-perf-meas)#exit
```

Verification: PM End-to-End Delay Management Configuration

This example shows how to use the **show performance-measurement summary** command to verify the PM for end-to-end delay management configuration.

```
R1#show performance-measurement summary
Total interfaces                              : 6
Total SR Policies                             : 1
Maximum PPS                                   : 1000 pkts/sec

SR Policy Delay-Measurement:
  Total sessions                              : 1
  Profile configuration:
    Measurement Type                          : Two-Way
    Computation interval                      : 30 seconds
    Burst interval                            : 3000 mSec
    Burst count                               : 10
    Protocol                                  : TWAMP-Lite Unauth
    HW Timestamp Supported                    : Yes
    Periodic advertisement                    : Enabled
      Interval                                : 30 (effective: 30) sec
      Threshold                               : 15%
      Minimum-Change                          : 600 uSec
```

```
    Accelerated advertisement                : Enabled
      Threshold                              : 25%
      Minimum-Change                         : 900 uSec
    Threshold crossing check                 : Minimum-delay
  Counters:
    Packets:
      Total sent                             : 334
      Total received                         : 0
    Errors:
      Total sent errors                      : 0
      Total received errors                  : 0
    Probes:
      Total started                          : 33
      Total completed                        : 0
      Total incomplete                       : 33
      Total advertisements                   : 0

Global Delay Counters:
  Total packets sent                         : 1251
  Total query packets received               : 917
  Total invalid session id                   : 0
  Total no session                           : 0

HW Support for MPLS-GAL [RFC6374] timestamp  : No
HW Support for TWAMP [RF5357] timestamp      : Yes
HW Support for 64 bit timestamp              : Yes
HW Support for IPv4 UDP Cheksum              : No
R1#
```

One-Way Link Loss Measurement

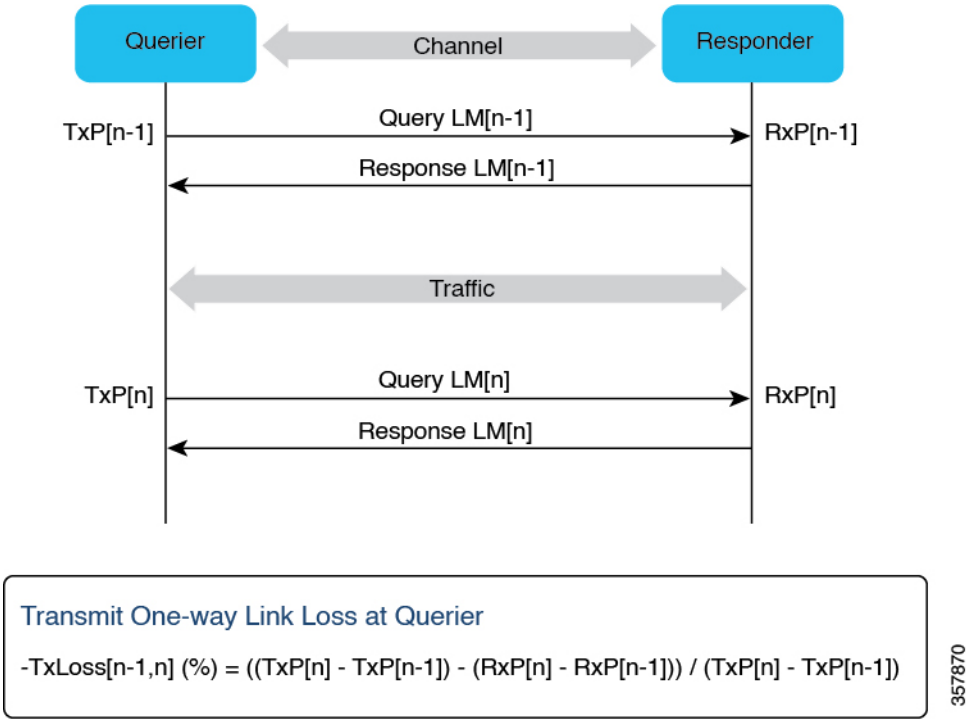
From Cisco IOS XE release 17.10.1a, a dual-color loss measurement mechanism is implemented to measure the link loss for point-to-point GRE-IPSec tunnels.

Information About One-Way Link Loss Measurement

Paths that are calculated across a network have to meet specified loss requirements to achieve specific SLAs. The One-Way Link Loss Measurement feature extends the existing network performance measurement capabilities to measure link loss and use it as a criterion to meet loss requirements of SLA.

To achieve this, an absolute one-way passive mechanism, which leverages the basic protocol of the *Simple Two-Way Direct Loss Measurement (SDLM)* over IP with User Datagram Protocol (UDP), is introduced.

Figure 35: Overview of One-Way Link Loss Measurement



Restrictions for One-Way Link Loss Measurement

- Only dual-color GRE one-way link loss measurement is supported.
- Dual-color loss measurement mechanism can only be used to measure point-to-point GRE-IPSec tunnel link loss.
- The Querier and Responder must use the same querier-dst-port UDP ports.
- The configured querier destination port (querier-dst-port) and querier source port (querier-src-port) must be different.
- The overlay destination IP address must be configured as next-hop for the measured GRE-IPSec tunnel.
- All measured interfaces must use the same GRE at both the Querier and the Responder ends.
- The maximum sessions supported are:
  - GRE-IPSec tunnel with BFD and IS-IS: 500
  - Performance measurement: 500
- Only IS-IS is supported as the Interior Gateway Protocol (IGP).

Supported Platforms for One-Way Link Loss Measurement

From Cisco IOS XE 17.10.1a, one-way link loss measurement is available on the following platforms:

- Cisco 1000 Series Aggregation Service Routers (ASR)

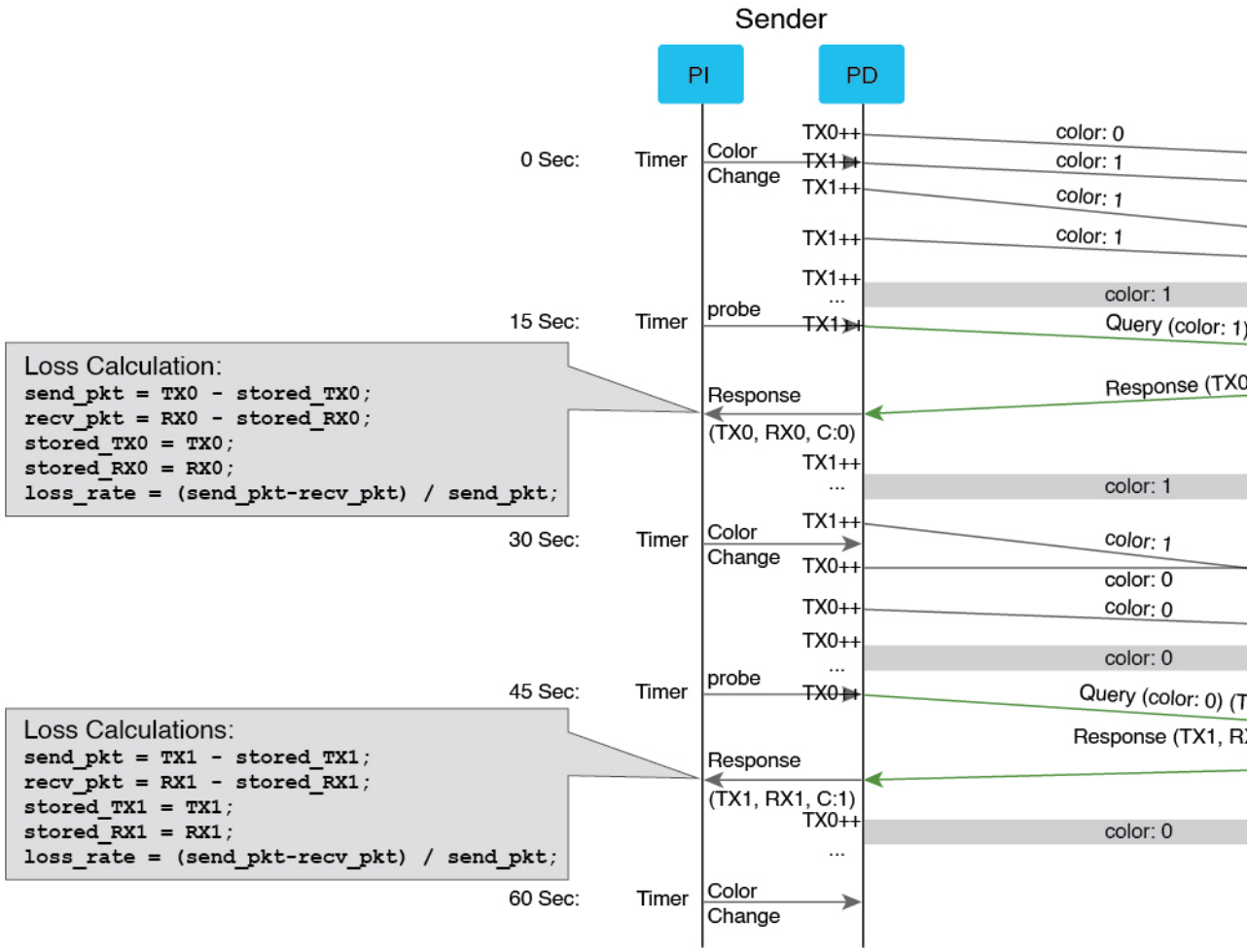
- Cisco Catalyst 8300 Series Edge Platforms
- Cisco Catalyst 8500 Series Edge Platforms
- Cisco Catalyst 8000V Edge Software

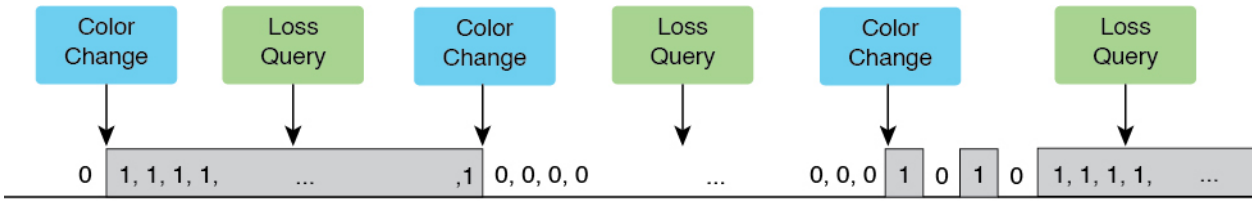
Dual-Color Loss Measurement for GRE-IPSec Tunnel

the dual-color loss-measurement mechanism is introduced to measure the GRE-IPSec tunnel link loss in real-time traffic. It uses one of the unused bits (9-12 in the **Flags** field) as the **Color** field of the GRE header (RFC 1701) for dual coloring. If not configured explicitly, the default value is 9.

A new keyword—**dual-color gre**—is introduced for the **color-type** command to implement the dual-color loss-measurement mechnism.

Figure 36: Dual-Color Loss Measurement Mechanism





The dual-color mechanism is implemented as follows:

- Traffic is tagged with Color value (0 or 1) in the Color field of the GRE header, and the Color value switches between 0 and 1 alternately at regular intervals.
- Colored traffic per color per interface is counted.
- The Unauthenticated SDLM format over IP with UDP protocol is used to encode the SDLM probe packet and SDLM reply probe packet. These packets carry the inactive TX or RX counters and calculate the loss at the querier side.

IGP IS-IS Advertisement for Link Loss Measurement

IGP advertises the extended traffic engineering link loss metric as a percentage, with or without the A-bit from Segment Routing performance measurement (SR-PM). No additional configuration is required in IS-IS to enable this advertisement.

- If the link loss value measured by SR-PM violates the configured *threshold* and *minimum-change* values, the SR-PM sends the value as a percentage to IS-IS, which advertises this value in the IS-IS domain.
- An **Anomalous (A) bit** is introduced to provide a new method to advertise the measured link loss value by checking the configured *lower-bound* and *upper-bound* values.
  - If the measured link loss value exceeds the *upper-bound* vaue and **A-bit** is not set, SR-PM sends the value as a percentage, with **A-bit** set, to IS-IS for advertisement.
  - If the measured link loss value falls below the *lower-bound* value and **A-bit** is set, SR-PM sends the value as a percentage, with **A-bit** unset, to IS-IS for advertisement.

IGP IS-IS Metric Penalty Option for Measured Link Loss

IS-IS implements the Metric Penalty mechanism by adding a new CLI under the ISIS interface to provide the option of increasing the IGP, TE, or both IGP and TE link metrics for the measured link when the A-bit is set, or decreasing it when the A-bit is unset:

isis metric fallback anomaly loss <*options*>

Configuration Example: One-Way Link Loss Measurement

Configuration at Querier

The following example shows the configuration at the querier side, with link loss measurement enabled for GRE-IPSec tunnel, and next-hop configured. The periodic interval for advertising the measured link loss value to IGP is set at 120 seconds, with the probe interval at 30 seconds. The lower and upper bounds for anomaly criteria are set at 0.5 and 1.0 for the default configuration, and at 1.0 and 2.0 for the sample configuration.

The default loss measurement probe color type is single color. In the following example, dual-color GRE is configured to enable the Loss Measurement functionality. IS-IS loss anomaly penalty can be set to either **Increment**, **Maximum**, or **Multiplier** options.

Default configuration:

```
loss-profile interfaces
  advertisement
   periodic
    interval 120
    threshold 10.000000
    minimum-change 0.100000
   anomaly-check
    lower-bound 0.500000 upper-bound 1.000000
   !
  probe
   tx-interval 30
   color-type
    dual-color gre
  !
```

Sample configuration:

```
performance-measurement
      protocol sdlm-light
        measurement loss
          unauthenticated
            querier-dst-port 6634
      dual-color gre-flags bit-position 9
      interface Tunnel55
        loss-measurement
          loss-profile name Profile1
      loss-profile name Profile1
        advertisement
          periodic
            interval 120
            threshold 10.0
            minimum-change 0.1
          anomaly-check
               lower-bound 1.0 upper-bound 2.0
        probe
          tx-interval 30
          color-type
            dual-color gre

interface Tunnel55
    ip address 10.0.0.10 10.255.255.0
    ip router isis 1
    mpls ip
    mpls traffic-eng tunnels
    tunnel source GigabitEthernet3
    tunnel destination 10.0.0.20
    tunnel protection ipsec profile gre_profile
    isis metric fallback anomaly loss maximum level-1
```

Configuration at Responder

The following example shows the configuration on the responder side:

```
performance-measurement
  protocol sdlm-light
   measurement loss
    unauthenticated
```

```
    querier-dst-port 6634
  dual-color gre-flags bit-position 9
```

Configuration Example: SR-MPLS Policy Configuration

The following example shows how to configure a static segment routing policy, and an on-demand segment routing policy:

```
segment-routing traffic-eng
 policy static-policy
  color 100 end-point 10.12.12.12
  candidate-paths
   preference 100
    constraints
     segments
      dataplane mpls
     !
    !
    dynamic
     metric
      type igp
     !
    !
   !
  !
 !
 on-demand color 100
  candidate-paths
   preference 100
    constraints
     segments
      dataplane mpls
     !
    !
    dynamic
     metric
      type igp
     !
    !
   !
  !
     !
```



Note You can configure either the static or the on-demand segment routing policy.

Verification: One-Way Link Loss Measurement

Use the **show performance-measurement summary** command on the querier side to provide information about the performance measurement parameters for link-loss measurement configuration:

```
show performance-measurement summary
Total interfaces                              : 1
Total SR Policies                             : 0
Total endpoints                               : 0
Maximum PPS                                   : 2000 pkts/sec
Dual-color gre bit-position                   : 9

Interface Delay-Measurement:
```

```
  Total sessions                            : 0
  Counters:
    Packets:
      Total sent                            : 0
      Total received                        : 0
    Errors:
      Total sent errors                     : 0
      Total received errors                 : 0
    Probes:
      Total started                         : 0
      Total completed                       : 0
      Total incomplete                      : 0
      Total advertisements                  : 0

SR Policy Delay-Measurement:
  Total sessions                            : 0
  Counters:
    Packets:
      Total sent                            : 0
      Total received                        : 0
    Errors:
      Total sent errors                     : 0
      Total received errors                 : 0
    Probes:
      Total started                         : 0
      Total completed                       : 0
      Total incomplete                      : 0
      Total advertisements                  : 0

Endpoint Delay-Measurement:
  Total sessions                            : 0
  Counters:
    Packets:
      Total sent                            : 0
      Total received                        : 0
    Errors:
      Total sent errors                     : 0
      Total received errors                 : 0
    Probes:
      Total started                         : 0
      Total completed                       : 0
      Total incomplete                      : 0
      Total advertisements                  : 0

Interface Loss-Measurement:
  Total sessions                            : 1
  Counters:
    Packets:
      Total sent                            : 22
      Total received                        : 10
    Errors:
      Total sent errors                     : 0
     Total received errors                  : 0
    Probes:
      Total started                         : 6
      Total completed                       : 2
      Total incomplete                      : 3
      Total advertisements                  : 6

Global Counters:
  Total packets sent                        : 22
  Total query packets received              : 10
  Total invalid session id                  : 0
  Total no session                          : 0
```

```
HW Support for MPLS-GAL [RFC6374] timestamp    : Yes
HW Support for IPv4 TWAMP [RF5357] timestamp   : Yes
HW Support for IPv6 TWAMP [RF5357] timestamp   : Yes
HW Support for 64 bit timestamp                : No
HW Support for IPv4 UDP Cheksum                : Yes
```

Use the **show performance-measurement sessions detail** command to provide detailed information about the performance measurement sessions for link-loss measurement configuration:

```
show performance-measurement sessions detail
Transport type              :Interface
Measurement type            :Loss Measurement
Interface name              :Tunnel100
Nexthop                     :100.0.0.2

Loss Measurement session:
  Session ID                :1
  Profile name              :loss1

  Last advertisement:
    Advertised at: 17:48:05 10-25 2022 (14 seconds ago)
    Advertised reason: First advertisement
    Advertised anomaly: INACTIVE
    Advertised loss(%) [Capped @ 50.331642%]: avg: 0.000000, min: 0.000000, max: 0.000000,
 variance: 0.000000

  Next advertisement:
    Check scheduled at the end of the current probe (roughly every 40 seconds)
    No probes completed
    Rolling average (%): 0.000000

  Current Probe:
    Started at 17:48:05 10-25 2022 (14 seconds ago)
    Packets Sent: 1, received: 1
    Measured loss(%) [Capped @ 50.331642%]: avg: 0.000000, min: 0.000000, max: 0.000000,
variance: 0.000000

Probe samples:
Rx Timestamp          Last TX     TX       Last RX     RX        Co Loss(0-100%)
17:48:10 10-25 2022   677         680       11         14        0  0.000000

Next probe scheduled at 17:48:45 10-25 2022 (in 26 seconds)
    Next burst packet will be sent in 1 seconds

  Liveness Detection:
    Session Creation Timestamp :10-25 17:32:00.699
    Session State: Up
    Last State Change Timestamp :10-25 17:47:40.761
    Missed count [consecutive] :0
    Received count [consecutive] :5
    Backoff                    :1
    Unique Path Name           :Path-1
    Loss in Last Interval      :0 % [TX: 1 RX: 1]
```

Use the **show performance-measurement profile loss interface** command on the querier side to view the performance measurement profile loss for interfaces:

```
show performance-measurement profile loss interface
Default Interface Loss Measurement:
  Profile configuration:
    Measurement Type                        : One-Way
    Tx interval                             : 10 sec
    Protocol                                : SDLM-Lite Unauth
    ToS DSCP value                          : 48
```

```
    Anomaly-check:
      lower-bound                           : 0.500000%
      upper-bound                           : 5.000000%
    Color-type:
      Dual-color:
        gre                                 : Enabled
    Periodic advertisement                  : Enabled
      Interval                              : 120 (effective: 120) sec
      Threshold                             : 15.000000%
      Minimum-Change                        : 0.200000
```

Use the **show performance-measurement interfaces name** <*name*> **detail** command on the querier side to view the performance measurement details, including link loss and delay, for a specific interface:

```
show performance-measurement interfaces name tunnel100 detail

sh performance-measurement interfaces name Tunnel100 det
Interface Name: Tunnel100 (ifh: 0x11)
  Delay-Measurement           : Disabled
  Loss-Measurement            : Enabled
  Local IPV4 Address          : 100.0.0.1
  Local IPV6 Address          : ::
  State                       : Up

  Loss Measurement session:
    Session ID                : 1
    Profile name              : Not configured

    Last advertisement:
      Advertised at: 10:23:40 10-25 2022 (32 seconds ago)
      Advertised reason: Periodic timer, avg loss threshold crossed
      Advertised anomaly: ACTIVE
     Advertised loss(%) [Capped @ 50.331642%]: avg: 9.458820, min: 9.997998, max: 10.002333,
 variance: 0.002499

    Next advertisement:
      Check scheduled at the end of the current probe (roughly every 40 seconds)
      No probes completed
      Rolling average (%): 9.458820

    Current Probe:
      Started at 10:23:40 10-25 2022 (32 seconds ago)
      Packets Sent: 3, received: 3
     Measured loss(%) [Capped @ 50.331642%]: avg: 6.667149, min: 0.000000, max: 10.002120,
 variance: 6.667149

      Probe samples:

Rx Timestamp          Last TX    TX       Last RX    RX        Col  Loss(0-100%)
10:24:05 10-25 2022   153911     153917   138520     138526    0    0.000000
10:23:55 10-25 2022   149505     177779   134556     160002    1    10.002120
10:23:45 10-25 2022   123899     153911   111509     138520    0    9.999333

      Next probe scheduled at 10:24:20 10-25 2022 (in 8 seconds)
      Next burst packet will be sent in 3 seconds

    Liveness Detection:
      Session Creation Timestamp: 10-25 10:09:56.898
      Session State: Up
      Last State Change Timestamp: 10-25 10:19:05.803
      Missed count   [consecutive]: 0
      Received count [consecutive]: 32
      Backoff                     : 1
      Unique Path Name            : Path-1
      Loss in Last Interval       : 0 % [TX: 3 RX: 3]
```

Use the **show performance-measurement history interfaces probe** command on the querier side to view the performance measurement probe history for the configured interface:

```
show performance-measurement history interfaces probe

Interface Name: Tunnel1 (ifh: 0x10)
 Loss-Measurement history (%):
    Session ID: 1
    Probe Start Timestamp     Pkt(TX/RX) Average   Min       Max
    23:28:56 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:28:16 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:27:36 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:26:56 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:26:16 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:25:36 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:24:56 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:24:16 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:23:36 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:22:56 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:22:16 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:21:36 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:20:56 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:20:16 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:19:36 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:18:56 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:18:16 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:17:36 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:16:56 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:16:16 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:15:36 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:14:56 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:14:16 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:13:36 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:12:56 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:12:16 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:11:36 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:10:56 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:10:16 08-04 2022       4/4        0.000000  0.000000  0.000000
    23:09:36 08-04 2022       4/4        0.000000  0.000000  0.000000
```

Use the **show isis teapp** command to view the IS-IS traffic engineering application information:

```
show isis teapp

Tag 200:
Tag 100:
Tag 1:
      ISIS TE Attr PM Information:
       Tu100: IDB num:14 Min:0 Max:0 Min-max-anomaly:0 Avg:0 Avg-anomaly:0 Var:0
 Is-Loss-set:1, Loss:533333 Loss-anomaly:1
       Tu200: IDB num:15 Min:0 Max:0 Min-max-anomaly:0 Avg:0 Avg-anomaly:0 Var:0
 Is-Loss-set:1, Loss:633333 Loss-anomaly:1
```

To view additional show commands, see Verification: PM Link Delay Configuration, on page 254.

Debugging and Troubleshooting One-Way Link Loss Measurement

- Use the show platform hardware qfp active interface if-name <*interface name*> | i PM command to check if INPUT_PM_DUAL_COLOR_LM (responder side) and OUTPUT_PM_DUAL_COLOR_LM (querier side) are enabled.

- Use the show platform hardware qfp active feature sr client grebit-pos command to check the GRE bit position.
- Use the show platform hardware qfp active feature sr client udp-ports command to check the source and destination UDP ports.
- Use the show platform hardware qfp active feature sr client dualcolor <*interface name*> command to check the current color.
- Use the following commands to clear performance measurement configuration and data:

```
clear performance-measurement
all                 clear all data
counters            clear pm querier counters
delay               clear pm querier delay
errors              clear internal errors
loss                clear pm querier loss
responder           clear responder data
```

- Use the following commands to debug performance measurement configuration:

```
debug performance-measurement
 all                 Performance Measurements all categories
 global              Global
 ha                  HA
 query               Query debugs
 responder           Responder debugs
```

Sample show Commands

```
R1#show performance-measurement interfaces detail
Interface Name: GigabitEthernet2 (ifh: 0x8)
Delay-Measurement : Enabled
Local IPV4 Address : 10.0.0.74
Local IPV6 Address : ::
State : Up
Delay Measurement session:
Session ID : 2
Last advertisement:
Advertised at: 06:45:50 02 2020 (214 seconds ago)
Advertised reason: First advertisement
Advertised delays (uSec): avg: 227, min: 198, max: 263, variance: 29
Next advertisement:
Check scheduled in 1 more probe (roughly every 160 seconds)
Aggregated delays (uSec): avg: 250, min: 208, max: 301, variance: 38
Rolling average (uSec): 254
Current Probe:
Started at 06:49:14 02 2020 (10 seconds ago)
Packets Sent: 3, received: 3
Measured delays (uSec): avg: 243, min: 230, max: 265, variance: 13
Probe samples:
Packet Rx Timestamp Measured Delay
06:49:22 02 2020 265500
06:49:18 02 2020 230000
06:49:14 02 2020 233500
Next probe scheduled at 06:49:54 02 2020 (in 30 seconds)
Next burst packet will be sent in 2 seconds
```

```
R1#show performance-measurement history interfaces name Gi2 probe
Interface Name: GigabitEthernet2 (ifh: 0x8)
  Delay-Measurement history (uSec):
```

```
        Probe Start Timestamp Pkt(TX/RX) Average   Min       Max
            06:48:34  02 2020 10/10      254       216       301
            06:47:54  02 2020 10/10      246       208       282
            06:47:14  02 2020 10/10      262       182       380
            06:46:34  02 2020 10/10      278       201       360
            06:45:54  02 2020 10/10      274       202       364
            06:45:14  02 2020 10/10      227       198       263

R1#show performance-measurement history interfaces name Gi2 aggr
Interface Name: GigabitEthernet2 (ifh: 0x8)
  Delay-Measurement history (uSec):
        Aggregation Timestamp Average   Min       Max       Action
            06:47:50  02 2020 259       182       380       NONE


R1#show performance-measurement counters interface name Gi2 detail
Interface Name: GigabitEthernet2 (ifh: 0x8)
  Delay-Measurement:
    Packets:
      Total sent                            : 67
      Total received                        : 67
    Errors:
        TX:
          Total interface down              : 0
          Total no MPLS caps                : 0
          Total no IP address               : 0
          Total other                       : 0
        RX:
          Total negative delay              : 0
          Total delay threshold exceeded    : 0
          Total missing TX timestamp        : 0
          Total missing RX timestamp        : 0
          Total probe full                  : 0
          Total probe not started           : 0
          Total control code error          : 0
          Total control code notif          : 0
    Probes:
      Total started                         : 6
      Total completed                       : 6
      Total incomplete                      : 0
      Total advertisements                  : 1


R1#show segment-routing traffic-eng policy all
Name: *10.2.2.2|100 (Color: 100 End-point: 10.2.2.2)
  Owners : BGP
  Status:
    Admin: up, Operational: up for 03:14:11 (since 12-02 03:36:05.290)
  Candidate-paths:
    Preference 100 (BGP):
      Dynamic (active)
        Metric Type: TE, Path Accumulated Metric: 30
          16002 [Prefix-SID, 10.2.2.2]
  Attributes:
    Binding SID: 40
      Allocation mode: dynamic
      State: Programmed
  IPv6 caps enabled


R1#show performance-measurement sr-policy name *10.2.2.2|100 detail
SR Policy name: *10.2.2.2|100
  Color                         : 100
  Endpoint                      : 10.2.2.2
  Source                        : 10.9.9.9
  Number of candidate-paths     : 1
```

```
  Candidate-Path:
    Preference                     : 100
    Protocol-origin                : BGP
    Discriminator                  : 0
    Active:                        : Yes
    Number of segment-lists        : 1
    Number of atomic paths         : 1
    Max Pkts per Burst             : 4000
    Max Pkts per Probe             : 40000
    AP Min Run per Probe           : 3
    Round-robin bursts             : 1
    Round-robin probes             : 1
    Last advertisement:
      Advertised at: 06:45:52  02 2020 (271 seconds ago)
      Advertised delays (uSec): avg: 860, min: 740, max: 946, variance: 120
    Next advertisement:
      Check scheduled in 1 more probe (roughly every 160 seconds)
      Aggregated delays (uSec): avg: 935, min: 795, max: 1146, variance: 140
    Last probe:
      Packets Sent: 10, received: 10
      Measured delays (uSec): avg: 910, min: 844, max: 1013, variance: 66
    Current Probe:
      Packets Sent: 8, received: 8
      Measured delays (uSec): avg: 949, min: 851, max: 1065, variance: 98

    Segment-List:
      Name                         : SegmentList0
      Number of atomic paths       : 1
      Last advertisement:
        Advertised at: 06:45:52  02 2020 (271 seconds ago)
        Advertised delays (uSec): avg: 860, min: 740, max: 946, variance: 120
      Next advertisement:
        Aggregated delays (uSec): avg: 935, min: 795, max: 1146, variance: 140
      Last probe:
        Packets Sent: 10, received: 10
        Measured delays (uSec): avg: 910, min: 844, max: 1013, variance: 66
      Current probe:
        Packets Sent: 8, received: 8
        Measured delays (uSec): avg: 949, min: 851, max: 1065, variance: 98

R1#show performance-measurement sr-policy name *10.2.2.2|100 private
SR Policy name: *10.2.2.2|100
  Color                            : 100
  Endpoint                         : 10.2.2.2
  Source                           : 10.9.9.9
  Number of candidate-paths        : 1

  Candidate-Path:
    Preference                     : 100
    Protocol-origin                : BGP
    Discriminator                  : 0
    Active:                        : Yes
    Number of segment-lists        : 1
    Number of atomic paths         : 1
    Max Pkts per Burst             : 4000
    Max Pkts per Probe             : 40000
    AP Min Run per Probe           : 3
    Round-robin bursts             : 1
    Round-robin probes             : 1
    Last advertisement:
      Advertised at: 06:45:52  02 2020 (284 seconds ago)
      Advertised delays (uSec): avg: 860, min: 740, max: 946, variance: 120
    Next advertisement:
```

```
      Check scheduled in 4 more probes (roughly every 160 seconds)
      Aggregated delays (uSec): avg: 935, min: 795, max: 1146, variance: 140
    Last probe:
      Packets Sent: 10, received: 10
      Measured delays (uSec): avg: 963, min: 851, max: 1083, variance: 112
    Current Probe:
      Packets Sent: 1, received: 1
      Measured delays (uSec): avg: 925, min: 925, max: 925, variance: 0

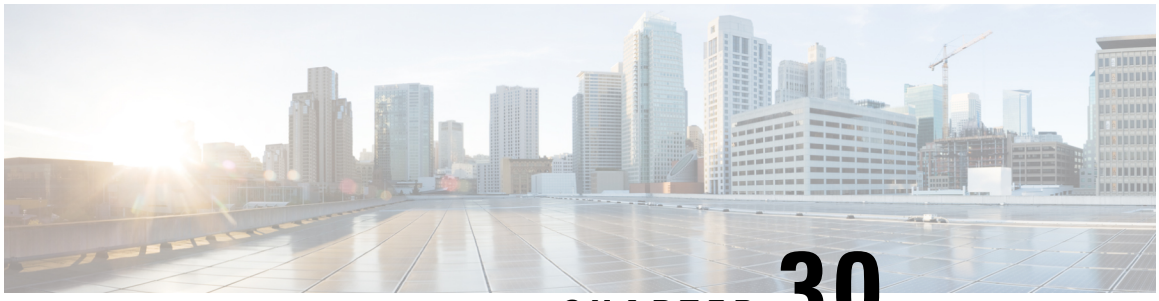
R1#show performance-measurement sr-policy name *10.2.2.2|100 verbose
SR Policy name: *10.2.2.2|100
  Color                          : 100
  Endpoint                       : 10.2.2.2
  Source                         : 10.9.9.9
  Number of candidate-paths      : 1

  Candidate-Path:
    Preference                   : 100
    Protocol-origin              : BGP
    Discriminator                : 0
    Active:                      : Yes
    Number of segment-lists      : 1
    Number of atomic paths       : 1
    Max Pkts per Burst           : 4000
    Max Pkts per Probe           : 40000
    AP Min Run per Probe         : 3
    Round-robin bursts           : 1
    Round-robin probes           : 1
    Last advertisement:
      Advertised at: 06:45:52  02 2020 (290 seconds ago)
      Advertised delays (uSec): avg: 860, min: 740, max: 946, variance: 120
    Next advertisement:
      Check scheduled in 4 more probes (roughly every 160 seconds)
      Aggregated delays (uSec): avg: 935, min: 795, max: 1146, variance: 140
    Last probe:
      Packets Sent: 10, received: 10
      Measured delays (uSec): avg: 963, min: 851, max: 1083, variance: 112
    Current Probe:
      Packets Sent: 3, received: 3
      Measured delays (uSec): avg: 911, min: 882, max: 925, variance: 29

PE3#show performance-measurement history sr-policy name *10.2.2.2|100  probe
SR Policy name: *10.2.2.2|100
  Candidate-Path:
    Preference               : 100
    Protocol-origin          : BGP
    Discriminator            : 0
    Active                   : Yes
        Probe Start Timestamp Pkt(TX/RX) Average   Min       Max
            06:49:54  02 2020 10/10      963       851       1083
            06:49:14  02 2020 10/10      910       844       1013
            06:48:34  02 2020 10/10      896       795       1019
            06:47:54  02 2020 10/10      1000      882       1146
            06:47:14  02 2020 10/10      990       909       1135
            06:46:34  02 2020 10/10      931       735       1080
            06:45:54  02 2020 10/10      911       768       1087
            06:45:14  02 2020 10/10      860       740       946
    Segment-list:
      Name                   : SegmentList0
          Probe Start Timestamp Pkt(TX/RX) Average   Min       Max
              06:49:54  02 2020 10/10      963       851       1083
              06:49:14  02 2020 10/10      910       844       1013
              06:48:34  02 2020 10/10      896       795       1019
              06:47:54  02 2020 10/10      1000      882       1146
              06:47:14  02 2020 10/10      990       909       1135
```

```
            06:46:34  02 2020 10/10      931       735       1080
            06:45:54  02 2020 10/10      911       768       1087
            06:45:14  02 2020 10/10      860       740       946
      Atomic path:
        Hops                  : 10.2.2.2
        Labels                : 16002
        Outgoing Interface    : GigabitEthernet2
        Next Hop              : 10.0.0.73
        Destination           : 10.2.2.2
        Session ID            : 1
            Probe Start Timestamp Pkt(TX/RX) Average   Min       Max
                06:49:54  02 2020 10/10      963       851       1083
                06:49:14  02 2020 10/10      910       844       1013
                06:48:34  02 2020 10/10      896       795       1019
                06:47:54  02 2020 10/10      1000      882       1146
                06:47:14  02 2020 10/10      990       909       1135
                06:46:34  02 2020 10/10      931       735       1080
                06:45:54  02 2020 10/10      911       768       1087
                06:45:14  02 2020 10/10      860       740       946

R1#show performance-measurement history sr-policy name *10.2.2.2|100  aggr
SR Policy name: *10.2.2.2|100
  Candidate-Path:
    Preference                : 100
    Protocol-origin           : BGP
    Discriminator             : 0
    Active                    : Yes
        Aggregation Timestamp Average   Min       Max       Action
            06:50:32  02 2020 942       795       1146      NONE
            06:47:52  02 2020 922       735       1135      NONE
    Segment-list:
      Name                    : SegmentList0
          Aggregation Timestamp Average   Min       Max       Action
              06:50:32  02 2020 942       795       1146      NONE
              06:47:52  02 2020 922       735       1135      NONE
      Atomic path:
        Hops                  : 10.2.2.2
        Labels                : 16002
        Outgoing Interface    : GigabitEthernet2
        Next Hop              : 10.0.0.73
        Destination           : 10.2.2.2
        Session ID            : 1
            Aggregation Timestamp Average   Min       Max       Action
                06:50:32  02 2020 942       795       1146      NONE
                06:47:52  02 2020 922       735       1135      NONE
```



CHAPTER 30

IP Endpoint Delay Measurement and Liveness Monitoring

This module describes the performance measurement for the IP Endpoint feature that measures the end-to-end delay and monitors liveness towards a specified IP endpoint.

- Information About IP Endpoint Performance Delay Measurement and Liveness Monitoring, on page 275
- Use Cases for IP Endpoint Performance Delay Measurement, on page 276
- How to Configure IP Endpoint Performance Delay Measurement, on page 278
- Configuration Examples for IP Endpoint Performance Delay Measurement, on page 283
- Verification for IP Endpoint Performance Delay Measurement, on page 285
- Feature Information for IP Endpoint Delay Measurement and Liveness Monitoring, on page 289

Information About IP Endpoint Performance Delay Measurement and Liveness Monitoring

The performance measurement for the IP Endpoint feature dynamically measures the end-to-end delay towards a specified IP endpoint. IP endpoints can be in the global routing table or VRFs.

Benefits of IP Endpoint Performance Delay Measurement and Liveness Monitoring

- Performance values (delay metrics and liveness states) are computed using the Two-Way Active Measurement Protocol (TWAMP) light.
- Support for TWAMP measurements using IP addresses in the global routing table, IPv4 VRFs, and IPv6 VRFs.
- Performance values, including histograms, are sent out using streaming telemetry, which is a push-based data collection technique, rather than a manual data collection technique.

Restrictions for IP Endpoint Performance Delay Measurement and Liveness Monitoring

- IP Endpoint Performance Delay Measurement with MPLS-Path is not supported.
- IP Endpoint Performance Delay Measurement with Loopback mode is not supported.
- The platform punt policer for TWAMP is recommended to configure for the scaling deployment.
- Be careful when changing the default burst interval (3 seconds) because this will directly influence the number of pps sent by the performance measurement feature.
- TWAMP Light support for both sender and responder.
- Two-way (must) and one-way (optional) delay measurements are supported. It is strongly recommended to use two-way delay measurement.
- One-way measurements should only be used for cases where the delay is expected to be in the order of tens/hundreds of milliseconds and the path is known to be asymmetric. The clocks on the source and target devices should be synchronized in this measurement.
- The use of Network Time Protocol (NTP) for clock synchronization is highly recommended.
- Minimum 2500 pps (250 probes/sessions each configured with 10 pps).
- Expected accuracy similar to IP-SLA.
- Configuration of TOS/DSCP for the TWAMP-light probes is supported.
- Configuration of packet size is supported.
- Altering padding size will increase CPU utilization.

Supported Platforms for IP Endpoint Performance Delay Measurement and Liveness Monitoring

The IP Endpoint Delay Measurement and Liveness Monitoring feature is available on the following platforms:

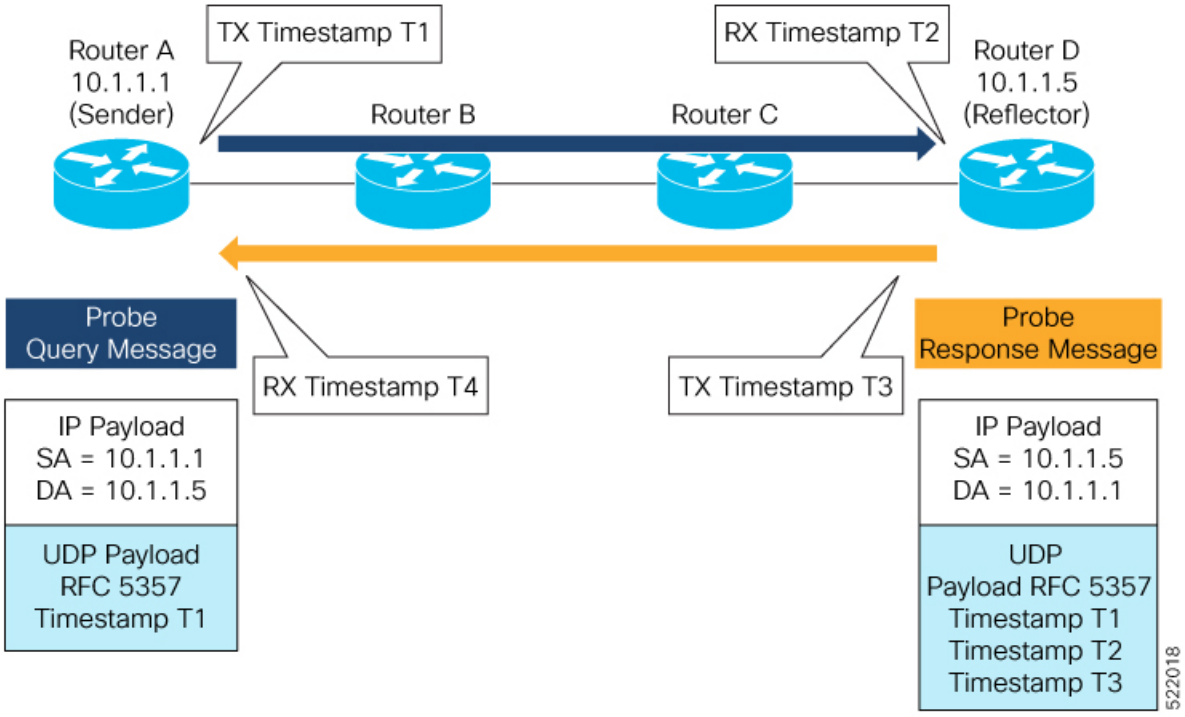
- Cisco Catalyst 8500 Series Edge Platforms
- Cisco Catalyst 8200 Series Edge Platforms
- Cisco ASR 1000 Series Aggregation Services Routers
- Cisco Catalyst 8000V Edge Software

Use Cases for IP Endpoint Performance Delay Measurement

The following use cases show different ways to deploy delay measurement and liveness detection for IP endpoints.

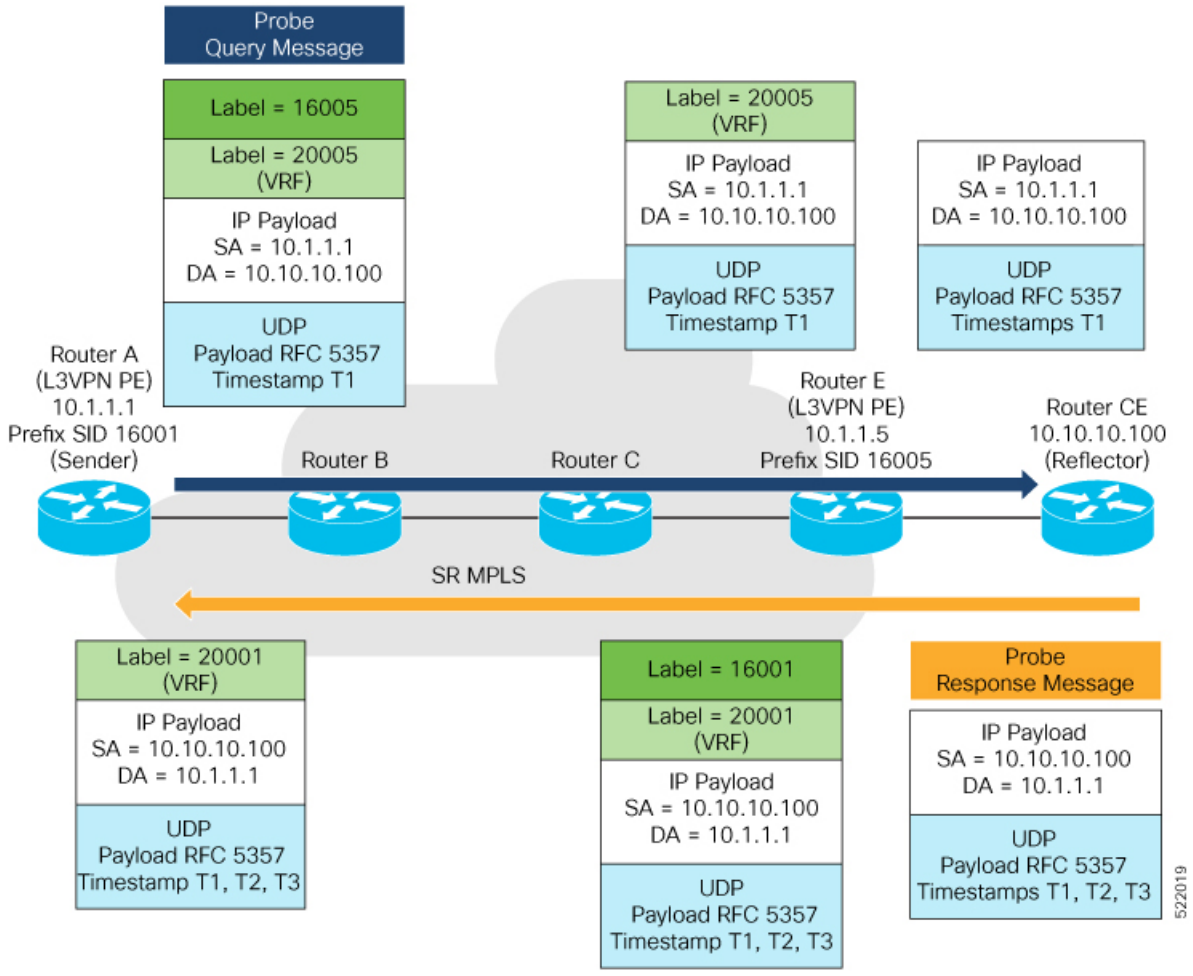
Use Case 1: Delay Measurement Probe Toward an IP Endpoint Reachable in the Global Routing Table

The following figure illustrates a delay measurement probe toward an IP endpoint reachable in the global routing table. The network interconnecting the sender and the reflector provides plain IP connectivity.



Use Case 2: Delay Measurement Probe Toward an IP Endpoint Reachable in a User-Specified VRF

The following figure illustrates a delay measurement probe toward an IP endpoint reachable in a user-specified L3VPN's VRF routing table. The L3VPN ingress PE (Router A) acts as the sender. The reflector is located in a CE device behind the L3VPN egress PE (Router E). The network interconnecting the L3VPN PEs provides MPLS connectivity with Segment Routing.



How to Configure IP Endpoint Performance Delay Measurement

Usage Guidelines and Limitations

- The endpoint of a probe is specified with an IP address. IPv4 and IPv6 endpoint addresses are supported.
- The endpoint of a probe can be any IP address reachable by the sender. For example, a local interface or a remote node or host located within an operator's network or reachable through a VRF.
- The endpoint's IP address can be located in the global routing table or under a user-specified VRF routing table.
- VRF-awareness allows operators to deploy probes in the following scenarios:
- Managed Customer Equipment (CE) scenarios:
  - PE to CE probes
  - CE to CE probes

- Unmanaged Customer Equipment (CE) scenarios:
  - PE to PE probes
  - PE to PE (source from PE-CE interface) probes
- SRv6 locator prefix and VRF SRv6 locator/function (uDT4/uDT6) as IPv6 endpoint of a probe is not supported.
- The endpoint's IP address can be reached through an IP path, MPLS LSP, or IP tunnel (GRE).
- When the endpoint is reachable using an MPLS LSP (for example, SR, LDP, RSVP-TE, SR Policy), the forwarding stage imposes the corresponding MPLS transport labels.
- When the endpoint is reachable via a VRF in an MPLS network, the forwarding stage imposes the corresponding MPLS service labels. In the forward path, the sender node uses the configured VRF for the endpoint address. In the return path, the reflector node derives the VRF based on which incoming VRF label the probe packet is received with.

Configuring Performance Delay Measurement IP Endpoint

Perform the following steps to configure the performance delay measurement IP endpoint.

SUMMARY STEPS

1. **enable**
2. **configureterminal**
3. **performance-measurement**
4. **endpoint** *endpoint-name*
5. **vrf** *vrf-name*
6. **source-address ipv4** *source_ip_address*
7. **destination-address ipv4** *destination_ip_address*
8. **delay-measurement**
9. **profile** *profile-name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**performance-measurement** **Example:** `RouterA(config)# performance-measurement`	Enters performance delay measurement mode.
Step 4	**endpoint** *endpoint-name* **Example:** `RouterA(config-perf-meas)# endpoint test-ipv4-1`	Specifies the name of the IP endpoint.
Step 5	**vrf** *vrf-name* **Example:** `RouterA(config-pm-ep)# vrf VPN-1`	Specifies the name of the VRF instance.
Step 6	**source-address ipv4** *source_ip_address* **Example:** `RouterA(config-pm-ep)# source-address ipv4 1.1.1.1`	Specifies the source IP address.
Step 7	**destination-address ipv4** *destination_ip_address* **Example:** `RouterA(config-pm-ep)# destination-address ipv4 1.1.1.4`	Specifies the destination IP address.
Step 8	**delay-measurement** **Example:** `RouterA(config-pm-ep)#delay-measurement`	Enables delay measurement on the IP endpoint.
Step 9	**profile** *profile-name* **Example:** `RouterA(config-pm-ep-dm)#profile test-profile`	Specifies the profile name.

Configuring IP Endpoint Performance Delay Measurement Profile

Perform the following steps to configure the IP endpoint performance delay measurement profile.

SUMMARY STEPS

1. **enable**
2. **configureterminal**
3. **performance-measurement**
4. **delay-profile endpoint name** *endpoint-name*
5. **probe**
6. **measurement-mode** {**one-way** | **two-way**}
7. **computation-interval** *seconds*
8. **burst-interval** *milliseconds*
9. **padding-size** *size*

10. **tos-dscp** *value*
11. **exit**
12. **liveness-detection**
13. **multiplier** *value*
14. **exit**
15. **advertisement**
16. **threshold-check** {**average-delay** | **maximum-delay** | **minimum-delay**}
17. **exit**
18. **periodic** {**disabled** | **interval** *seconds* | **minimum-change** *microseconds* | **threshold** *value*}
19. **exit**
20. **accelerated** {**minimum-change** *microseconds* | **threshold** *value*}
21. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**performance-measurement** **Example:** `RouterA(config)# performance-measurement`	Enters performance delay measurement mode.
Step 4	**delay-profile endpoint name** *endpoint-name* **Example:** `RouterA(config-perf-meas)#delay-profile endpoint name test-profile`	Specifies the delay profile name to be associated.
Step 5	**probe** **Example:** `RouterA(config-pm-dm-ep)#probe`	Enters probe configuration mode.
Step 6	**measurement-mode** {**one-way**	**two-way**} **Example:** `RouterA(config-pm-dm-ep-probe)# measurement-mode two-way`	Specifies the interval measurement mode. The default value is two-way measurement mode.
Step 7	**computation-interval** *seconds* **Example:**	Specifies the interval for metric computation. The range is from 1 to 3600 seconds. The default value is 30 seconds.

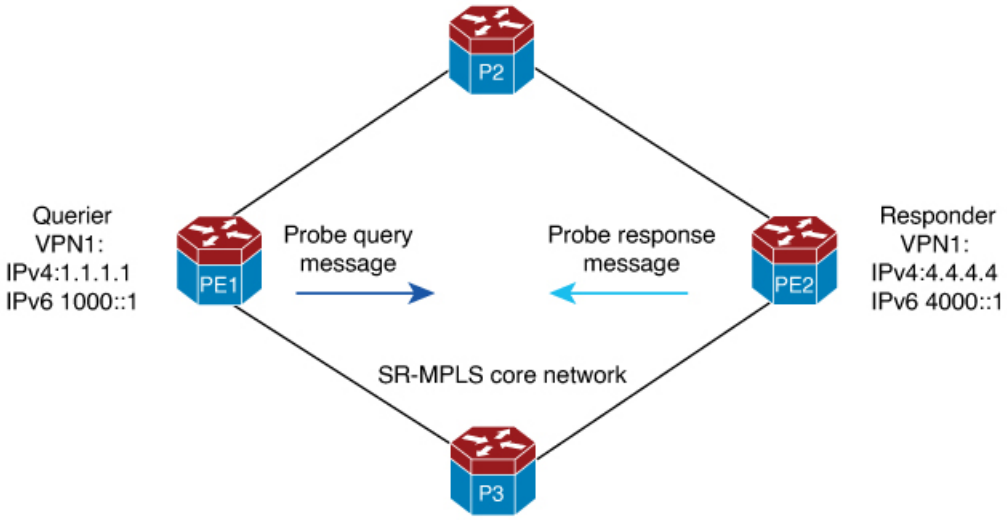
	Command or Action	Purpose
	`RouterA(config-pm-dm-ep-probe)# computation-interval 60`	
Step 8	**burst-interval** *milliseconds* **Example:** `RouterA(config-pm-dm-ep-probe)#burst-interval 3000`	Specifies the interval for sending probe packet. The range is from 30 to 15000 milliseconds. The default value is 3000 milliseconds.
Step 9	**padding-size** *size* **Example:** `RouterA(config-pm-dm-ep-probe)#padding-size 0`	Specifies the packet pading size. The range is from 0 to 8000. **Note** It is not recommended to configure padding fragmentation and reassembly which impac
Step 10	**tos-dscp** *value* **Example:** `RouterA(config-pm-dm-ep-probe)# tos-dscp 48`	Specifies the type of Service DSCP. The range is from 0 to 63.The default value is DSCP 48 for IP/UDP.
Step 11	**exit** **Example:** `RouterA(config-pm-dm-ep-probe)#exit`	Exits probe configuration mode.
Step 12	**liveness-detection** **Example:** `RouterA(config-pm-dm-ep)#liveness-detection`	Enters endpoint liveness detection configuration mode.
Step 13	**multiplier** *value* **Example:** `RouterA(config-pm-dm-ep-live)#multiplier 3`	Specify the number of probe packets sent before the head-end node assumes the endpoint liveness session state is down. The range is from 2 to 10.
Step 14	**exit** **Example:** `RouterA(config-pm-dm-ep-live)#exit`	Exits endpoint liveness detection configuration mode.
Step 15	**advertisement** **Example:** `RouterA(config-pm-dm-ep)#advertisement`	Enters advertisement configuration mode.
Step 16	**threshold-check** {**average-delay** | **maximum-delay** | **minimum-delay**} **Example:** `RouterA(config-pm-dm-ep-adv)#threshold-check maximum-delay`	Enters threshold check advertisement configuration mode. Checks the delay metric change for threshold crossing for accelerated advertisement. The default value is average-delay.

	Command or Action	Purpose
Step 17	**exit** **Example:** `RouterA(config-pm-dm-ep-adv-threshold-check)#exit`	Exits threshold check advertisement configuration mode.
Step 18	**periodic** {**disabled** | **interval** *seconds* | **minimum-change** *microseconds* | **threshold** *value*} **Example:** `RouterA(config-pm-dm-ep-adv)# periodic interval 120`	Enters periodic advertisement configuration mode. **disabled**: Disables periodic advertisement. **interval**: Periodic advertisement and metric aggregation interval. The range is from 30 to 3600 seconds. The default value is 120 seconds. **minimum-change**: Periodic advertisement minimum change value. The range is from 0 to 1000000 microseconds. The default value is 500 microseconds. **threshold**: Specifies the minimum-delay metric change for threshold crossing for periodic advertisement. The range is from 0 to 100 percent. The default value is 10 percent. **Note** An advertisement happens when both n
Step 19	**exit** **Example:** `RouterA(config-pm-dm-ep-adv-per)#exit`	Exits periodic advertisement configuration mode.
Step 20	**accelerated** {**minimum-change** *microseconds* | **threshold** *value*} **Example:** `RouterA(config-pm-dm-ep-adv)#accelerated minimum-change 1000`	Enters accelerated advertisement configuration mode. **minimum-change**: Periodic advertisement minimum change value. The range is from 1 to 1000000 microseconds.. The default value is 500 microseconds. **threshold**: Specifies the minimum-delay metric change for threshold crossing for periodic advertisement. The range is from 0 to 100 percent. The default value is 20 percent.
Step 21	**exit** **Example:** `RouterA(config-pm-dm-ep-adv-acc)#exit`	Exits accelerated advertisement configuration mode.

Configuration Examples for IP Endpoint Performance Delay Measurement

The following are configuration examples for the IP endpoint performance delay measurement.

Configuration Example: Configuring IP Endpoint Performance Delay Measurement (global configuration)



Running Configuration

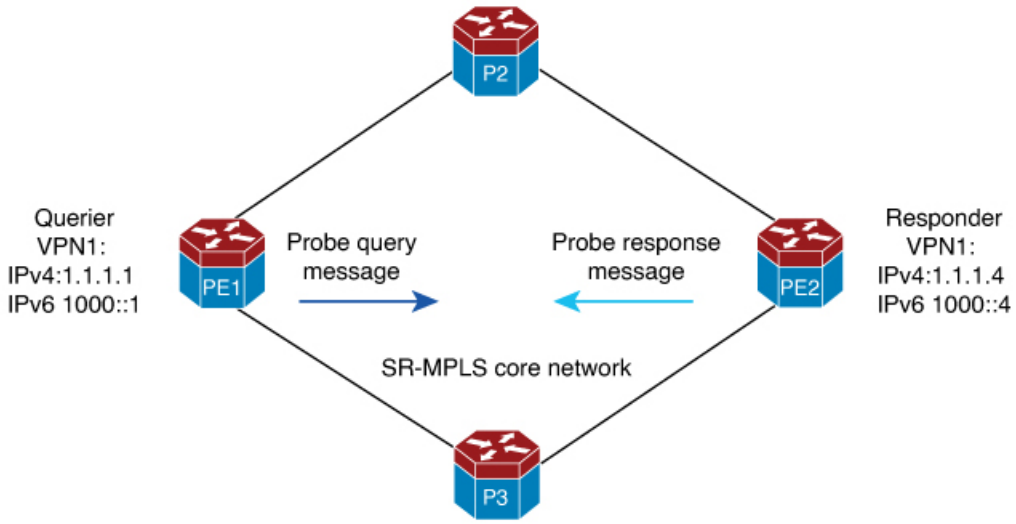
Querier (Sender) configuration:

```
performance-measurement
delay-profile endpoint name test-profile
probe
burst-interval 95
tos-dscp 24
!
endpoint test-ipv4-1
source-address ipv4 1.1.1.1
destination-address ipv4 4.4.4.4
delay-measurement
profile test-profile
performance-measurement
!
endpoint test-ipv6-1
source-address ipv6 1000::1
destination-address ipv6 4000::1
delay-measurement
profile test-profile
```

Querier (Sender) configuration:

```
performance-measurement
```

Configuration Example: Configuring IP Endpoint Performance Delay Measurement (VRF configuration)



Running Configuration

Querier (Sender) configuration:

```
performance-measurement
delay-profile endpoint name test-profile
probe
burst-interval 95
tos-dscp 24
!
endpoint test-ipv4-1
vrf VPN-1
source-address ipv4 1.1.1.1
destination-address ipv4 1.1.1.4
delay-measurement
profile test-profile
!
performance-measurement
!
endpoint test-ipv6-1
vrf VPN-1
source-address ipv6 1000::1
destination-address ipv6 1000::4
delay-measurement
profile test-profile
```

Querier (Sender) configuration:

```
performance-measurement
```

Verification for IP Endpoint Performance Delay Measurement

You can use the following show commands for verifying the IP endpoint performance delay measurement.

- **show performance-measurement counters endpoint filter name pm-name detail**

- **show performance-measurement endpoint filter name pm-name detail**
- **show performance-measurement history endpoint adv**
- **show performance-measurement history endpoint aggr**
- **show performance-measurement history endpoint filter name liveness-notification**
- **show performance-measurement history endpoint filter name pm-name adv**
- **show performance-measurement history endpoint filter name pm-name aggr**
- **show performance-measurement history endpoint filter name probe**
- **show performance-measurement profile endpoint**
- **show performance-measurement responder counters interface**
- **show performance-measurement responder summary**
- **show performance-measurement summary**

Examples

The following are sample outputs of the show commands for verifying the IP endpoint performance delay measurement.

show performance-measurement summary

```
pe1#show performance-measurement summary
Total interfaces : 1
Total SR Policies : 0
Total endpoints : 250
Maximum PPS : 2000 pkts/sec
Dual-color gre bit-position : 9
Endpoint Delay-Measurement:
    Total sessions : 250
    Counters:
        Packets:
            Total sent : 12816719
            Total received : 11443853
        Errors:
            Total sent errors : 0
            Total received errors : 172421
        Probes:
            Total started : 40959
            Total completed : 35208
            Total incomplete : 5751
            Total advertisements : 33
```

show performance-measurement profile

```
pe1#show performance-measurement profile endpoint name test-profile
test-profile Endpoint Delay Measurement:
    Profile configuration:
        Measurement Type : Two-Way
        Computation interval : 30 (effective : 30) seconds
        Burst interval : 95 mSec
        Burst count : 316
        Protocol : TWAMP-Lite Unauth
```

```
    ToS DSCP value : 48
    Destination sweeping mode : Disabled
    Periodic advertisement : Enabled
        Interval : 120 (effective: 120) sec
        Threshold : 10%
        Minimum-Change : 500 uSec
    Accelerated advertisement : Disabled
    Threshold crossing check : Average-delay
    Liveness-detection multiplier : 3
```

show performance-measurement endpoint session

```
PE#show performance-measurement endpoint filter name test-ipv6-2000 detail
Endpoint name: test-ipv6-2000
  Source address                  : 1412::1
  Destination address             : 1412::4
  Delay Measurement               : Enabled
  VRF                             : Not configured
  Profile name                    : test-profile
  Forward SID List                : Not configured
  Reverse SID List                : Not configured

  Delay Measurement session:
    Session ID               : 6
    Profile name             : test-profile

    Last advertisement:
      Advertised at: 15:00:52 11-01 2023 (30 seconds ago)
      Advertised reason: Periodic timer, avg delay threshold crossed
      Advertised anomaly: INACTIVE
      Advertised delays (uSec): avg: 4265, min: 2902, max: 5999, variance: 986

    Next advertisement:
      Check scheduled in 3 more probes (roughly every 120 seconds)
      No probes completed
      Rolling average (uSec): 4378

    Current Probe:
      Started at 15:00:52 11-01 2023 (30 seconds ago)
      Packets Sent: 311, received: 311
      Measured delays (uSec): avg: 5004, min: 4010, max: 5977, variance: 994
      Probe samples:
        Packet Rx Timestamp       Measured Delay (nsec)
        15:01:23 11-01 2023       5093944
        15:01:23 11-01 2023       5092502
        15:01:23 11-01 2023       5110069
        15:01:23 11-01 2023       5365350
        15:01:23 11-01 2023       5365940
      Next probe scheduled at 15:01:22 11-01 2023 (in 0 seconds)
      Burst completed

    Liveness Detection:
      Session Creation Timestamp: 11-01 14:50:29.937
      Session State: Up
      Last State Change Timestamp: 11-01 14:51:56.110
      Missed count   [consecutive]: 0
      Received count [consecutive]: 5691
      Backoff                     : 0
      Unique Path Name            : Path-6
      Loss in Last Interval       : 0 % [TX: 313 RX: 313]
```

show performance-measurement counters endpoint

```
pe1#show performance-measurement counters endpoint filter name test-ipv6-100 detail
Endpoint name: test-ipv6-100
Source address : 1000::100
Destination address : 6000::100
Delay Measurement : Enabled
VRF : Not configured
Profile name : test-profile
Forward SID List : Not configured
Reverse SID List : Not configured
Delay-Measurement:
    Packets:
        Total sent : 771804
        Total received : 741831
    Errors:
        TX:
            Total interface down : 0
            Total no MPLS caps : 0
            Total no IP address : 0
            Total other : 0
        RX:
            Total negative delay : 1762
            Total delay threshold exceeded : 0
            Total missing TX timestamp : 0
            Total missing RX timestamp : 0
            Total probe full : 0
            Total probe not started : 1
            Total control code error : 0
            Total control code notif : 0
        Probes:
            Total started : 2487
            Total completed : 2358
            Total incomplete : 128
            Total advertisements : 1
```

show performance-measurement history endpoint

```
pe1#show performance-measurement history endpoint filter name test-ipv6-100 probe
Endpoint name: test-ipv6-100
Source address : 1000::100
Destination address : 6000::100
Delay Measurement : Enabled
VRF : Not configured
Profile name : test-profile
Forward SID List : Not configured
Reverse SID List : Not configured
Delay-Measurement history (uSec):
Probe Start Timestamp     Pkt(TX/RX)     Average       Min     Max
03:52:25 01-18 2024     313/313             30         16      314
03:51:54 01-18 2024     313/313             29         13      47
03:51:22 01-18 2024     313/313             30         15      349
03:38:16 01-18 2024     10/10               42         29      101
<snip>
03:37:46 01-18 2024     10/10               37         28       45
03:37:16 01-18 2024     10/10               37         29       47
03:36:46 01-18 2024     10/10               38         31       46
03:36:16 01-18 2024     10/10               39         28       48
```

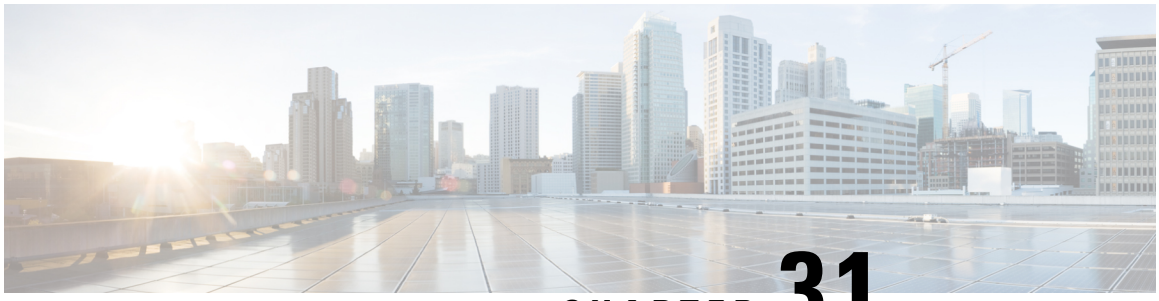
Feature Information for IP Endpoint Delay Measurement and Liveness Monitoring

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 38: Feature Information for IP Endpoint Delay Measurement and Liveness Monitoring

Feature Name	Releases	Feature Information
IP Endpoint Delay Measurement and Liveness Monitoring	IOS XE 17.14.1a	This feature enables you to measure the end-to-end delay and monitor liveness towards either a specified IPv4 or IPv6 endpoint. This feature is introduced for the following platforms: • Cisco Catalyst 8500 Series Edge Platforms • Cisco Catalyst 8200 Series Edge Platforms • Cisco ASR 1000 Series Aggregation Services Routers • Cisco Catalyst 8000V Edge Software From Cisco IOS XE 17.14.1a, you can be configure this feature using the **performance-measurement endpoint** and **performance-measurement delay-profile endpoint** commands.



CHAPTER 31

SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated)

This chapter describes how Segment Routing Traffic Engineering (SR-TE) works with the Per-flow policy (PFP) On-Demand Next-hop (ODN) and auto steering (Per flow ODN/AS) mechanism. This chapter contains the following sections:

- Feature Information for SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated), on page 291
- Information About SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated), on page 292
- BGP Color Extended Community and VRF Prefix Coloring, on page 293
- Support for PFP with RIB Path, on page 296
- Configuring SR-TE Per-Flow Class (ODN) and Automated Steering (PCE Delegated), on page 297
- Verifying SR-TE Per-Flow Class (ODN) and Automated Steering (PCE Delegated), on page 299

Feature Information for SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated)

Table 39: Feature History

Feature Name	Release	Description
Support for PFP with RIB Path	Cisco IOS XE 17.9.1a	This feature enables you to configure forwarding class in a per-flow policy using the Routing Information Base (RIB) path option. Instead of configuring a per-destination policy, the RIB option uses the IGP shortest path to the policy destination.

Feature Name	Release	Description
Attaching Extended Color Communities to BGP VRF	Cisco IOS XE 17.7.1a Cisco IOS XE 17.11.1a	This feature introduces new methods of attaching extended color communities to a prefix. A color community is an indicator of the bandwidth or latency level of the traffic being sent to the prefix. The following are the new ways of attaching them to the prefix: • VRF export coloring • VRF import coloring • Route redistribution coloring in BGP • Neighbor inbound coloring From Cisco IOS XE 17.11.1a, this feature is extended to the following platforms: • Cisco Catalyst 8300 Series Edge Platforms • Cisco Catalyst 8500 Series Edge Platforms • Cisco Catalyst 8000V Edge Software
SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated)	Cisco IOS XE Amsterdam 17.4	This feature lets you steer traffic with SR-TE PFP based on the QoS markings on the packets. The traffic is then switched onto the appropriate path based on the forward classes of the packet.

Information About SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated)

The Segment Routing-Traffic Engineering (SR-TE) Per-Flow policy (PFP) On-Demand Next-Hop (ODN) with auto-steering (Per-Flow ODN/AS) is a mechanism that allows the steering of traffic on a segment routing policy based on the attributes of the packets. SR-TE PFP ODN with auto steering (Per flow ODN/AS) is a mechanism that allows the steering of traffic on an SR policy based on the attributes of the packets. Packets are classified using Cisco's Modular QoS CLI (MQC) framework and then marked using internal tags known as Forward Classes (FCs). A PFP is then used to route the marked packets based on the mappings between an FC and its corresponding path. This means that the traffic is steered based on its QoS markings and switched to the appropriate path based on the FC of the packet.

A PFP is identified by <*color, endpoint*>. It is configured with a per-flow forwarding class table with up to eight entries, with each entry indexed by an FC and pointing to a Per Destination Policy (PDP).

Note The following features are supported:

- 250 PFP+PDP (Combination)
- 6 PE and 6 VPE
- 10k VPNV4 prefix limit
- L3VPN Inter AS Option B for SR PFP
- IPv6 over PFP

Restrictions for SR-TE Per-Flow (Class) ODN and Automated Steering (PCE Delegated)

- Dynamic change in the Quality of Service (QoS) policy is not supported.
- PIC core over SR-TE tunnel PIC edge is not supported.
- VPLS over SR-TE is not supported.
- Configure the set forward class to 0 to take the default path for nonforward class.
- BGP Labeled Unicast (BGP-LU) (RFC 3107) is not supported for SR ODN PFP Auto Steering.
- L2VPN over PFP tunnels is not supported.
- Performance measurement over PFP is not supported.
- MPLS ping or trace route over PFP is not supported.
- Auto route announcement over PFP or PDP is not supported.
- PIC is not supported over PFP.

BGP Color Extended Community and VRF Prefix Coloring

In the Segment Routing Traffic Engineering mechanism, the prefix that needs an SR-TE routing path is associated with a color-extended community (an attribute that assigns color to the prefixes). Currently, BGP has the capability to attach the color-extended community based only on the neighbor command routemap outbound configuration. To color the prefixes based on attributes such as Source-VRF, Destination-VRF, CE-neighbor, and Source protocol, the following ways of attaching color are introduced:

- VRF Export Coloring
- VRF Import Coloring
- Route Redistribution Coloring into BGP

- Neighbor In-bound Coloring

Additionally, in Cisco IOS XE releases prior to 17.7.1a, any new color-extended community attached to the prefix replaces the existing color-extended community available in the prefix. To be able to add the new color-extended community to the existing list of color-extended communities instead of replacing, the keyword **additive** is added to the **route-map** command as part of Cisco IOS XE 17.7.1a:

```
route-map SRTE-color-map permit
set extcommunity color < 1-4294967295> [additive]
```



Note When a BGP update is received with multiple color-extended communities, the highest color value in the list is used for SR policy creation, and the binding SID corresponding to the SR policy is used as the routing path for the received BGP path. If the SR policy corresponding to the highest color is not available, BGP uses the interface as the routing path for the update.

Supported Platforms

From Cisco IOS XE 17.7.1a, this feature is supported on:

- Cisco ASR 1000 Series platforms

From Cisco IOS XE 17.11.1a, this feature is supported on:

- Cisco Catalyst 8300 Series Edge platforms
- Cisco Catalyst 8500 Series Edge platforms
- Cisco Catalyst 8000V Edge software

Attaching a Color-Extended Community

The following ways of attaching color-extended communities are available:

- VRF Export Coloring: The following configuration attaches a color extended community to the VPN prefix as per the export route map color-extended community associated with the VRF. This enables the association of the color-extended community based on the source VRF of the VPN prefix:

```
route-map SRTE-color-map permit
set extcommunity color < 1-4294967295> [additive]
vrf def SRTE-VRF
rd 1:1
!
address-family ipv4
export map SRTE-color-map
exit-address-family
!
address-family ipv6
export map SRTE-color-map
exit-address-family
```

- VRF Import Coloring: The following configuration attaches a color-extended community to an imported VRF prefix as per the import route map color-extended community associated with the VRF. This enables the attachment of the color-extended community to a prefix based on the VRF the prefix is imported to:

```
route-map SRTE-color-map permit
set extcommunity color < 1-4294967295> [additive]
vrf def SRTE-VRF
rd 1:1
!
address-family ipv4
import map SRTE-color-map
exit-address-family
!
address-family ipv6
import map SRTE-color-map
exit-address-family
```

- Route Redistribution Coloring into BGP: The following configuration attaches a color-extended community as part of the redistribution routes to BGP. This associates the color-extended community to a prefix based on the source protocol owning the prefix:

```
route-map SRTE-color-map permit
set extcommunity color < 1-4294967295> [additive]
router bgp <ASnum>
address-family ipv4
redistribute <source-protocol> route-map SRTE-color-map
or
network <address> mask <network-mask> route-map SRTE-color-map
exit-address-family
!
address-family ipv4 vrf <vrf-name>
redistribute <source-protocol> route-map SRTE-color-map
or
network <address> mask <network-mask> route-map SRTE-color-map
exit-address-family
!
address-family ipv6
redistribute <source-protocol> route-map SRTE-color-map
or
network <address>/masklen route-map SRTE-color-map
exit-address-family
!
address-family ipv6 vrf <vrf-name>
redistribute <source-protocol=> route-map SRTE-color-map
or
network <address>/masklen route-map SRTE-color-map
exit-address-family
```

- Neighbor Inbound Coloring: The following configuration attaches a color-extended community as part of the inbound route map processing attached to the neighbor. This attaches a color-extended community based on the neighbor advertising the prefix:

```
route-map SRTE-color-map permit
set extcommunity color < 1-4294967295> [additive]
router bgp <ASnum>
address-family ipv4
neighbor <address> route-map SRTE-color-map in
exit-address-family
!
address-family vpnv4
neighbor <address> route-map SRTE-color-map in
exit-address-family
!
address-family ipv4 vrf <vrf-name>
neighbor <address> route-map SRTE-color-map in
exit-address-family
!
address-family ipv6
```

```
neighbor <address> route-map SRTE-color-map in
exit-address-family
!
address-family vpnv6
neighbor <address> route-map SRTE-color-map in
exit-address-family
!
address-family ipv6 vrf <vrf-name>
neighbor <address> route-map SRTE-color-map in
exit-address-family
```

Support for PFP with RIB Path

PFP consists of a bundle output chain element (OCE), and each hash of the bundle OCE consists of a PDP policy (PDP tunnel). In this scenario, a PDP policy is created for the default IGP or RIB learned path. This means that a separate PDP policy is created for every default IGP or RIB learned path. Therefore, this implementation will eventually increase the number of policies and will not scale.

From Cisco IOS XE 17.9.1a, you can configure forwarding class in a PFP using the RIB path option. Instead of configuring a PDP, the RIB option uses the IGP shortest path to the policy destination.

PFP has a binding SID, similar to the PDP. The traffic-steering mechanism is also the same as PDP, either through BSID or through RIB.

A PFP is in the operational UP state based on the following conditions:

- The default FC is configured with a PDP, and it is in the operational UP state.
- The default FC is configured with the RIB path and is resolved.

Note The state of the nondefault FC does not affect the PFP state.

After a packet is steered on the PFP, according to the FC marked by Modular QoS CLI (MQC) at ingress, the following scenarios show the path of the packet:

- If PFP is in the Down state, the packet is dropped.
- If no FC is attached to a packet, the packet is forwarded with a default FC in PFP.
- If an FC is attached to a packet that points to a resolved RIB path or an operational PDP, the packet is forwarded to it.
- If an FC attached on a packet points to a nonexisting unresolved RIB path or a nonoperational PDP, the packet is forwarded to the default FC.

Example: Configuring PFP with RIB Path

The following example shows how to configure PFP using both the RIB path and color:

```
segment-routing traffic-eng
  policy PERFLOW
   color 10 end-point 1.1.1.1
```

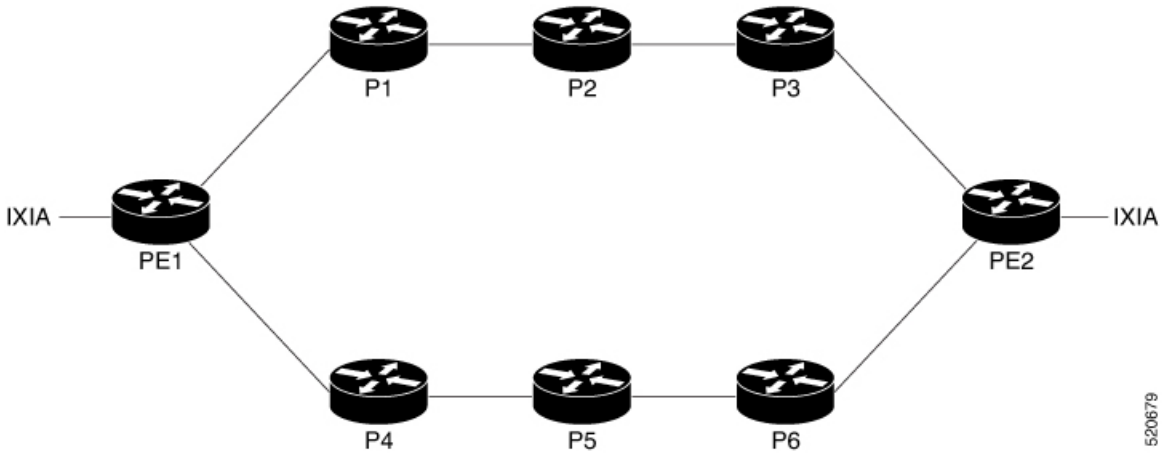
```
binding-sid mpls 15001
candidate-path
 preference 1
  per-flow
   forward-class 0 rib
   forward-class 1 color 20
   forward-class 2 color 30
```

The following example shows how to configure the ODN PFP using both the RIB path and color:

```
segment-routing traffic-eng
 on-demand color 10
  candidate-path
   preference 1
    per-flow
     forward-class 0 rib
     forward-class 1 color 20
     forward-class 2 color 30
```

Configuring SR-TE Per-Flow Class (ODN) and Automated Steering (PCE Delegated)

Consider the following topology:



Perform the following steps to configure ODN for PFP:

1. Configure QOS on PE1:

```
class-map DSCP
        match DSCP AF41
```

   - Set the forward class on the class map:

     ```
     policy-map type epbr per-flow
      class DSCP
      set forward-class 1
     ```

   - Attach the policy map on the corresponding interface:

     ```
     interface GigabitEthernet0/0/3
      service-policy type epbr input PFP
     ```

2. Configure SR-TE PFP on PE1:

   - Set the forward class on PFP:

```
on-demand color 4500
 authorized
candidate-paths
  preference 2
    per-flow
        forward-class 0 color 100
        forward-class 0 rib
        forward-class 2 color 102
```

   - Attach the segment list to PDP:

```
policy  perflow_pdp
color 100 end-point 10.5.5.5
candidate-paths
 preference 2
  explicit segment-list srte1 weight 10
  !
  constraints
    segments
      dataplane mpls
```

   - Set the segment list to SR-TE:

```
segment-routing traffic-eng
  segment-list name srte1
       index 1 mpls label 16002
       index 2 mpls label 16005
```

3. Configure SR-TE PFP on PE2:

```
ip prefix-list pfp seq 5 permit 10.35.0.0/16 le 32
```

   - Attach the route map to PFP:

```
route-map pfp permit 10
 match ip address prefix-list pfp
 set extcommunity color 4500
```

   - Activate the BGP routes:

```
router bgp 100
!
 address-family vpnv4
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 send-community extended
  neighbor 10.1.1.1 route-map pfp out
```

4. View the PFP output:

```
show segment-routing traffic-eng policy name *6.6.6.6|4090 detail

Name: *6.6.6.6|4090 (Color: 4090 End-point: 6.6.6.6)
Owners : BGP
Status:
Admin: up, Operational: up for 01:29:41 (since 06-21 14:09:05.510)
Candidate-paths:
Preference 1 (BGP):
Per-flow Information (active):
Forward PDP PDP BSID RW
Class Color Status Status
-------- ---------- ------- --------
```

```
0 rib n/a n/a
1 129 up Done
2 130 up Done
3 131 up Done
4 132 up Done
Default Forward Class: 0
Attributes:
Binding SID: 39
Allocation mode: dynamic
State: Programmed
IPv6 caps enabled
Tunnel ID: 65568 (Interface Handle: 0x26)
Per owner configs:
BGP
Binding SID: dynamic
Stats:
5 minute output rate 0 bits/sec, 0 packets/sec
Packets: 500524 Bytes: 88056352

Event history:
Timestamp Client Event type Context: Value
--------- ------ ---------- -------: -----
06-21 14:09:05.489 BGP Policy created Name: BGP
06-21 14:09:05.490 BGP Set colour Colour: 4090
06-21 14:09:05.490 BGP Set end point End-point: 6.6.6.6
06-21 14:09:05.490 BGP Set dynamic pce Path option: per flow
06-21 14:09:05.510 BGP BSID allocated FWD: label 39
06-21 14:09:05.510 FH Resolution Policy state UP Status: PFP RESOLVED CP: 1
06-21 14:09:05.551 FH Resolution REOPT triggered Status: REOPTIMIZED CP: 1
06-21 14:09:05.576 FH Resolution REOPT triggered Status: REOPTIMIZED CP: 1
06-21 14:09:05.602 FH Resolution REOPT triggered Status: REOPTIMIZED CP: 1
06-21 14:09:05.626 FH Resolution REOPT triggered Status: REOPTIMIZED CP: 1
```

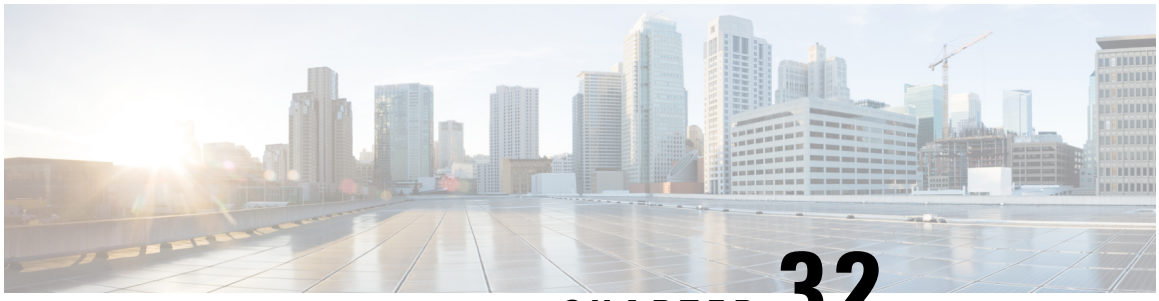
Verifying SR-TE Per-Flow Class (ODN) and Automated Steering (PCE Delegated)

Use the following command to verify SR-TE Per-Flow Class (ODN) and Automated Steering (PCE Delegated):

show segment-routing traffic-eng policy name *10.5.5.5|4500

```
Name: *10.5.5.5|4500 (Color: 4500 End-point: 10.5.5.5)
Owners : BGP
Status:
Admin: up, Operational: up for 00:03:50 (since 09-07 16:07:02.938)
Candidate-paths:
Preference 2 (BGP):
Per-flow Information (active):
Forward PDP PDP BSID RW
Class Color Status Status
-------- ---------- ------- --------
0 100 up Done
1 101 up unknown Pending
2 102 up unknown Pending
Default Forward Class: 0
Attributes:
Binding SID: 72
Allocation mode: dynamic
State: Programmed
IPv6 caps enabled
Tunnel ID: 65675 (Interface Handle: 0x2D)
```

```
Per owner configs:
BGP
Binding SID: dynamic
Stats:
5 minute output rate 0 bits/sec, 0 packets/sec
Packets: 9 Bytes: 584
```



CHAPTER 32

Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains

A border router can advertise the same loopback interface prefixes and the associated prefix Segment Identifiers (SIDs) in multiple ISIS domains.

- Feature Information for Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains, on page 301
- Information about Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains, on page 302
- How to Configure Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains, on page 302
- Example: Configure Loopback Prefix SIDs of a BR in Multiple ISIS Domains, on page 304

Feature Information for Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains

Table 40: Feature Information for Performance Measurement for Traffic Engineering

Feature Name	Releases	Feature Information
Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains	Cisco IOS XE Amsterdam 17.3.2	A border router can advertise loopback interface prefixes and the associated prefix Segment Identifiers (SIDs) in multiple ISIS domains. With such an advertisement, the routers in each associated domain can communicate with the border router using the same prefixes and prefix SIDs.

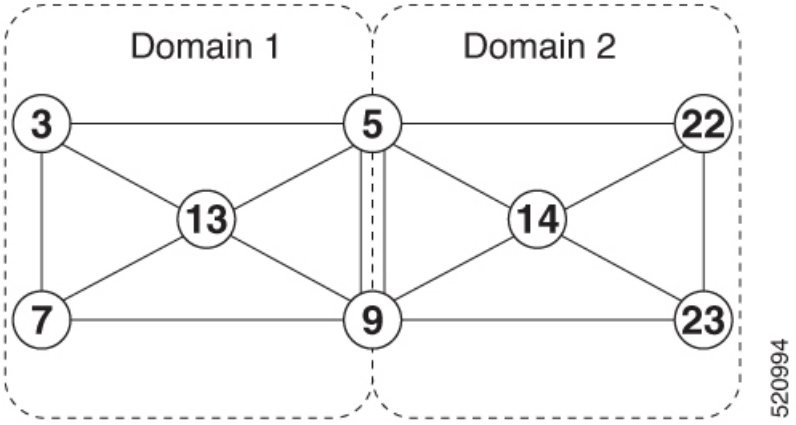
Information about Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains

Overview of the Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains

In a segment routing deployment having multiple ISIS domains, it would be beneficial if a border router advertises loopback interface prefixes and prefix SIDs in each associated domain. With such an advertisement, the routers in each associated domain can communicate with the border router using the same prefixes and prefix SIDs.

This feature provides a border router with the capability to advertise prefixes and prefix SIDs into multiple ISIS routing processes, and thereby, into each associated domain.

For example, in the topology shown in the following diagram, the border routers, Router 5 and Router 9, can advertise their prefixes and prefix SIDs in both Domain 1 and Domain 2. A router in Domain 1, say Router 3, and a router in Domain 2, say Router 22, can use the same prefix SIDs to send traffic to send traffic to either border router.



How to Configure Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains

Configure the Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains

To advertise a loopback prefix and the prefix SID of a border route in multiple ISIS domains, on the border router, issue the **passive-interface** *loopback-interface-name* command to the ISIS routing process for each domain.

```
router isis 1
  passive-interface loopback 0
router isis 2
  passive-interface loopback 0
```

Verify the Advertisement of Loopback Prefix SIDs of a Border Router in Multiple ISIS Domains

```
Router#show isis database verbose

Tag 1:
IS-IS Level-1 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime/Rcvd      ATT/P/OL
Router.00-00        * 0x00000013   0xDCD8                469/*         0/0/0
  Area Address: 49.0001
  NLPID:        0xCC
  Router CAP:   10.0.0.0, D:0, S:0
    Segment Routing: I:1 V:0, SRGB Base: 16000 Range: 8000
    Segment Routing Local Block: SRLB Base: 15000 Range: 1000
    Segment Routing Algorithms: SPF, Strict-SPF
    Node-MSD
      MSD: 16
  Hostname: Router
  Metric: 0          IP 10.2.2.2/32
    Prefix-attr: X:0 R:0 N:0
  Metric: 0          IP 10.1.1.1/32
    Prefix-attr: X:0 R:0 N:0
    Prefix-SID Index: 1, Algorithm:SPF, R:0 N:1 P:0 E:0 V:0 L:0
IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime/Rcvd      ATT/P/OL
Router.00-00        * 0x00000014   0xDAD9                469/*         0/0/0
  Area Address: 49.0001
  NLPID:        0xCC
  Router CAP:   10.0.0.0, D:0, S:0
    Segment Routing: I:1 V:0, SRGB Base: 16000 Range: 8000
    Segment Routing Local Block: SRLB Base: 15000 Range: 1000
    Segment Routing Algorithms: SPF, Strict-SPF
    Node-MSD
      MSD: 16
  Hostname: Router
  Metric: 0          IP 10.2.2.2/32
    Prefix-attr: X:0 R:0 N:0
  Metric: 0          IP 10.1.1.1/32
    Prefix-attr: X:0 R:0 N:0
    Prefix-SID Index: 1, Algorithm:SPF, R:0 N:1 P:0 E:0 V:0 L:0

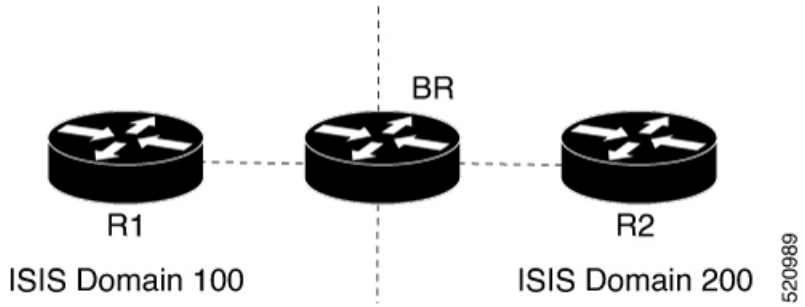
Tag 2:
IS-IS Level-1 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime/Rcvd      ATT/P/OL
Router.00-00        * 0x00000012   0xC68A               1179/*         0/0/0
  Area Address: 39.0002
  NLPID:        0xCC
  Router CAP:   10.1.1.1, D:0, S:0
    Segment Routing: I:1 V:0, SRGB Base: 16000 Range: 8000
    Segment Routing Local Block: SRLB Base: 15000 Range: 1000
    Segment Routing Algorithms: SPF, Strict-SPF
    Node-MSD
      MSD: 16
  Hostname: Router
  IP Address:   10.1.1.1
  Metric: 0          IP 10.1.1.1/32
    Prefix-attr: X:0 R:0 N:1
```

```
    Prefix-SID Index: 1, Algorithm:SPF, R:0 N:1 P:0 E:0 V:0 L:0
IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime/Rcvd      ATT/P/OL
Router.00-00        * 0x00000011   0xC889               1184/*         0/0/0
  Area Address: 39.0002
  NLPID:        0xCC
  Router CAP:   10.1.1.1, D:0, S:0
    Segment Routing: I:1 V:0, SRGB Base: 16000 Range: 8000
    Segment Routing Local Block: SRLB Base: 15000 Range: 1000
    Segment Routing Algorithms: SPF, Strict-SPF
    Node-MSD
      MSD: 16
  Hostname: Router
  IP Address:   10.1.1.1
  Metric: 0          IP 10.1.1.1/32
    Prefix-attr: X:0 R:0 N:1
    Prefix-SID Index: 1, Algorithm:SPF, R:0 N:1 P:0 E:0 V:0 L:0
```

Example: Configure Loopback Prefix SIDs of a BR in Multiple ISIS Domains

The following example shows how to configure a BR and the association of a prefix SID in multiple domains.

Consider the following topology in which we have routers R1 and R2 in two different ISIS domains, and a border router BR that belongs to both the domains.

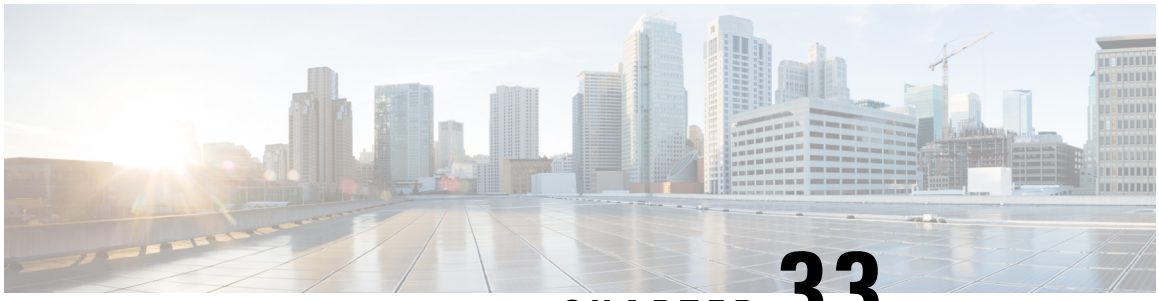


Device	Loopback Address	Prefix SID
R1	10.1.1.1/32	101
R2	10.2.2.2/32	202
BR	10.3.3.3/32	303

The following configuration on the border router BR causes the router to advertise its loopback interface address and the associated prefix SID in both the connected ISIS domains. This configuration example shows the definition of a loopback interface, the association of a prefix SID with the loopback interface, and the advertisement of the loopback interface address and the associated prefix SID in the ISIS domains ISIS 100 and ISIS 200.

```
BR>enable
BR#configure terminal
BR(config)#interface loopback 0
BR(config-if)#ip address 10.3.3.3 255.255.255.255
BR(config-if)#exit
```

```
BR(config)#segment-routing mpls
BR(config-srmpls)#connected-prefix-sid-map
BR(config-srmpls-conn)#address-family ipv4
BR(config-srmpls-conn-af)#10.3.3.3/32 index 303 range 1
BR(config-srmpls-conn-af)#exit-address-family
BR(config-srmpls-conn-af)#end
BR#configure terminal
BR(config)#router isis 100
BR(config-router)#passive-interface loopback 0
BR(config-router)#exit
BR(config)#router isis 200
BR(config-router)#passive-interface loopback 0
BR(config-router)#end
```



CHAPTER 33

Traffic Steering by Dropping Invalid Paths

If the SR-TE policy has no valid paths defined, the paths are dropped, and the traffic that is being steered through the policy falls back to the default (unconstrained IGP) forwarding path. Also, when an SR-TE policy carrying best-effort traffic fails, the traffic is re-routed and which in turn impacts the SLA(service level agreements) for premium traffic.

To solve the issue of SR-TE policy failing, the traffic in the data plane is dropped but kept in the control plane. Therefore, other segment routing policies, which could be potentially be carrying premium traffic, are not impacted.

- Feature Information for Traffic Steering by Dropping Invalid Paths, on page 307
- How to Configure Traffic Steering by Dropping Invalid Paths , on page 308

Feature Information for Traffic Steering by Dropping Invalid Paths

Thefollowing table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

UseCisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 41: Feature Information for Performance Measurement for Traffic Engineering

Feature Name	Releases	Featuure Information
Traffic Steering By Dropping Invalid Paths	Cisco IOS XE Bengaluru 17.5	If the SR-TE policy fails, the traffic in the data plane is dropped but kept in the control plane. Therefore, other segment routing policies, potentially carrying premium traffic, are not impacted.

Overview

If the SR-TE policy has no valid paths defined, the paths are dropped, and the traffic that is being steered through the policy falls back to the default (unconstrained IGP) forwarding path. Also, when an SR-TE policy carrying best-effort traffic fails, the traffic is re-routed and which in turn impacts the SLA(service level agreements) for premium traffic.

To solve the issue of SR-TE policy failing, the traffic in the data plane is dropped but kept in the control plane. Therefore, other segment routing policies, which could be potentially be carrying premium traffic, are not impacted

This feature can be configured by using the **path-invalidation drop** command.

Before You Begin

This feature should not be enabled if you have already configured segment routing BFD or performance liveness monitoring. If this feature is enabled, segment routing BFD or performance liveness notification is ignored. In such a scenario, no logging or syslog notification is generated for segment routing BFD or performance liveness events.

Note that if the SR-TE policy is in Down state and this feature is configured, the state of the SR-TE policy is not affected.

Benefits

- Configuring this feature ensures that other segment routing policies that are configured to route premium traffic are not impacted thereby ensuring that SLA guidelines are not affected.

Restrictions

- This feature cannot be enabled in combination with segment routing BFD or performance monitoring liveness check.

How to Configure Traffic Steering by Dropping Invalid Paths

Configuring for a PCC Profile

This configuration results in a PCE-initiated policy having the path-invalidation functionality enabled for a policy instantiated with a profile ID matching the configured value:

```
segment-routing traffic-eng
  pcc
   profile <number >
    steering
     path-invalidation drop
```

Configuring for Static Policies

This configuration results in a configuring path validation drop for a segment routing static policy:

```
segment-routing traffic-eng
  policy <name>
   steering
    path-invalidation drop
```

Configuring for On-Demand Next Hop for SR-TE Policies

This configuration results in a configuring path validation drop for an on-demand segment routing policy for a specific color:

```
segment-routing traffic-eng
  on-demand color <>
   steering
    path-invalidation drop
```

Show Commands

Use the **show segment-routing traffic-eng policy name** command to view path invalidation event types and invalidation drop status.

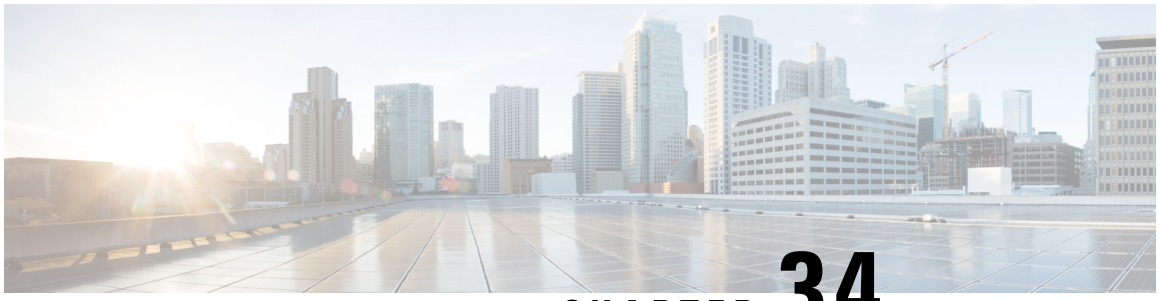
```
device#show segment-routing traffic-eng policy name foo detail
  Name: foo (Color: 10 End-point: 192.168.0.8)
    Owners : CLI
    Status:
      Admin: up, Operational: up for 00:00:08 (since 09-17 10:19:54.536)
    Candidate-paths:
      Preference 100 (CLI):
        Dynamic (active)
          Status: Invalidation drop
          Metric Type: TE
    Attributes:
      Binding SID: 20
        Allocation mode: dynamic
        State: Programmed
      Autoroute:
        Include all
    Tunnel ID: 65536 (Interface Handle: 0x9)
    Per owner configs:
      CLI
        Binding SID: dynamic
    Stats:
      5 minute output rate 0 bits/sec, 0 packets/sec
      Packets: 0   Bytes: 0

    Event history:
      Timestamp            Client              Event type              Context: Value
      ---------            ------              ----------              -------: -----
      09-17 10:19:54.536   CLI                Policy created           Name: CLI
      09-17 10:19:54.537   CLI                Path Invalidation        Drop: Configured
      09-17 10:19:58.744   CLI                Set colour               Colour: 10
      09-17 10:19:58.744   CLI               Set end point           End-point: 192.168.0.8

      09-17 10:19:58.752   CLI                Set dynamic              Path option: dynamic

      09-17 10:19:58.753   CLI                BSID allocated           FWD: label 20
      09-17 10:19:58.755   FH Resolution      Policy state UP         Status: PATH RESOLVED
```

```
CP: 100
     09-17 10:19:58.760   FH Resolution      REOPT triggered          Status: REOPTIMIZED
CP: 100
     09-17 10:19:58.780   CLI                Path Invalidation        Drop: Unconfigured
     09-17 10:19:59.537   CLI                Path Invalidation        Drop: Set
     09-17 10:20:00.853   FH Resolution      Path Invalidation        Status: Drop
     09-17 10:20:01.853   FH Resolution      Path Invalidation        Status: No Drop
     09-17 10:20:02.853   FH Resolution      Path Invalidation        Status: Drop
```



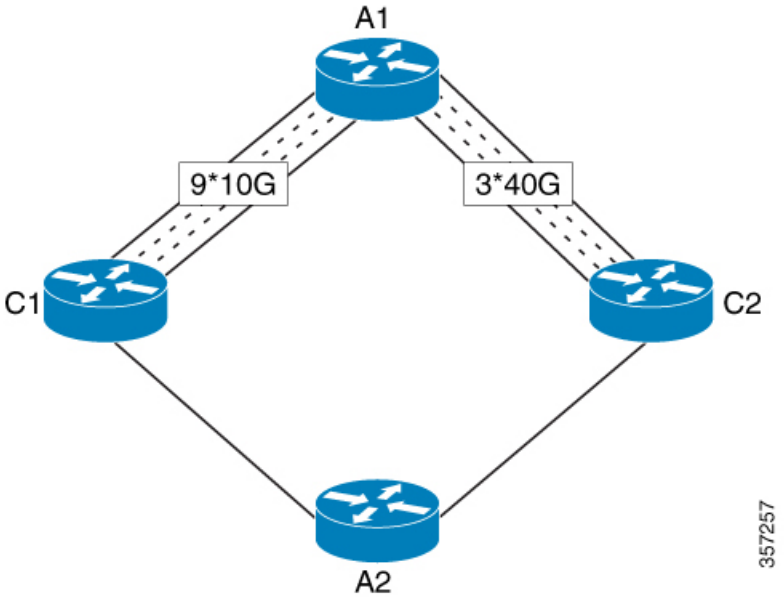
CHAPTER 34

Configuring the Cisco ISIS Local Unequal Cost Multipath (UCMP)

The Cisco IOS XE ISIS Local UCMP feature allows you to load balance traffic from A1 to A2, across all the links from A1-C1 and A1-C2 in a network. When you configure equal metrics on all the links, it will create Equal Cost Multipath (ECMP) paths. However, the higher bandwidth links will carry the same traffic as the lower bandwidth links and the higher bandwidth links are underutilized. To avoid this problem, you can configure all the links to distribute the traffic proportionately across the links based on bandwidth, even if the configured metrics on all links are the same.

The following figure explains the topology:

Figure 37: Local Unequal Cost Multipath Topology



- Configuring the Unequal Cost Multi Path (UCMP) Local , on page 312
- Verifying the Unequal Cost Multi Path (UCMP) Local, on page 312
- Debug Commands, on page 313
- Feature Information for Segment Routing—IS-IS UCMP, on page 313

Configuring the Unequal Cost Multi Path (UCMP) Local

Perform the following task to configure the ucmp local:

```
router isis
 ucmp local [prefix-list <prefix-list-name>]
router isis
 address-family ipv6
 ucmp local [prefix-list <prefix-list-name>]
```

Verifying the Unequal Cost Multi Path (UCMP) Local

To verify the feature, use the following show commands:

- **show interface <name> counters**
- **show ip route**
- **show ipv6 route**
- **show ip cef**
- **show mpls forwarding-table labels detail**
- **show mpls infrastructure lfd lte**

Examples: Show Commands

The following is a sample output from the show ip route of the Unequal Cost Multi Path (UCMP) Local:

```
Device#show ip route 10.138.1.3
Routing entry for 10.138.1.0/24
Known via "isis", distance 115, metric 50, type level-1
Redistributing via isis Ring#1
Advertised by isis Ring#1 (self originated)
Last update from 10.148.1.1on FortyGigabitEthernet0/5/1, 00:24:51
ago
Routing Descriptor Blocks:
* 10.198.1.1, from 10.1.1.1, 00:24:51 ago, via GigabitEthernet0/0/0
 Route metric is 50, traffic share count is 6
 10.148.1.1, from 10.1.1.1, 00:24:51 ago, via
FortyGigabitEthernet0/5/1
 Route metric is 50, traffic share count is 25
```

Note You should verify if the *traffic share count* is computed according to the interface bandwidth.

The following is a sample output from show interface counter of the Unequal Cost Multi Path (UCMP) Local:

```
Device#show interface fo0/5/1 counters
Port      InOctets   InUcastPkts InMcastPkts InBcastPkts
Fo0/5/1   22883         0           17           0
```

```
Port      OutOctets  OutUcastPkts OutMcastPkts OutBcastPkts
Fo0/5/1   16242883     57513        17           0
PE12#show interface gi0/0/0 counters
Port      InOctets  InUcastPkts InMcastPkts InBcastPkts
Gi0/0/0    26388        26          19           0
Port      OutOctets  OutUcastPkts  OutMcastPkts OutBcastPkts
Gi0/0/0    81944464    264216       195          0
```

Note You can verify if the outgoing traffic is split according to the computed traffic share count.

Debug Commands

To troubleshoot the issues related to local UCMP, use the following debug commands:

- **debug isis mfi**
- **debug ip routing detail**
- **debug ipv6 routing**

Feature Information for Segment Routing—IS-IS UCMP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 42: Feature Information for Segment Routing—IS-IS UCMP

Feature Name	Releases	Feature Information
Segment Routing—IS-IS UCMP	Cisco IOS XE 17.5.1	The Segment Routing—IS-IS UCMP feature allows you to load balance outgoing traffic across all IGP ECMP paths proportionally to the interface bandwidth.



CHAPTER 35

Enabling Segment Routing Flexible Algorithm

Segment Routing Flexible Algorithm allows operators to customize IGP shortest path computation according to their own needs. An operator can assign custom SR prefix-SIDs to realize forwarding beyond link-cost-based SPF. As a result, Flexible Algorithm provides a traffic engineered path automatically computed by the IGP to any destination reachable by the IGP.

The SR architecture associates prefix-SIDs to an algorithm which defines how the path is computed. Flexible Algorithm allows for user-defined algorithms where the IGP computes paths based on a user-defined combination of metric type and constraint.

- Feature History, on page 316
- Prerequisites for Flexible Algorithm, on page 317
- Restrictions for Flexible Algorithm, on page 317
- Building Blocks of Segment Routing Flexible Algorithm, on page 317
- Flexible Algorithm Prefix-SID Redistribution , on page 319
- Flexible Algorithm Prefix Metric Advertisement, on page 320
- Flexible Algorithm Configurations, on page 321
- Verifying the Flexible Algorithm Configuration, on page 327

Feature History

Table 43: Feature History

Feature Name	Release Information	Feature Description
TE Metric Support for IS-IS Flexible Algorithm	Cisco IOS XE Dublin 17.11.1a	Flexible algorithm allows user-defined algorithms where the Interior Gateway Protocol (IGP) computes paths based on a user-defined combination of metric type (path optimization objective) and constraint. This feature adds support for the TE metric as a metric type for the IS-IS Flexible Algorithm feature. This allows the TE metric along with the IGP and delay metrics to be used when running the shortest path computations.
Segment Routing Flexible Algorithm Prefix SID Redistribution	Cisco IOS XE Cupertino 17.8.1	With this feature, prefix SIDs are provided for all supported algorithms when a prefix is redistributed. This feature is enabled automatically when you configure redistribution of routes with strict or Flexible Algorithm SIDs.
IS-IS Flexible Algorithm Include Affinity Support	Cisco IOS XE Bengaluru 17.6.1	This feature supports **include-any** and **include-all** affinities in IS-IS. Prior to Cisco IOS XE Bengaluru 17.6.1 release, only Flexible Algorithm affinity **exclude-any** was supported.
Segment Routing Flexible Algorithm	Cisco IOS XE Bengaluru 17.4.1	TI LFA and uLoop Avoidance: Allows computation of Loop Free Alternate (LFA) paths. TI-LFA backup paths using the same constraints as the calculation of the primary paths for Flexible Algorithms, for IS-IS.

Feature Name	Release Information	Feature Description
Segment Routing Flexible Algorithm	Cisco IOS XE Amsterdam17.3.1	Segment Routing Flexible Algorithm allows operators to customize IGP shortest path computation according to their own needs. An operator can assign custom SR prefix-SIDs to realize forwarding beyond link-cost-based SPF. As a result, Flexible Algorithm provides a traffic engineered path automatically computed by the IGP to any destination reachable by the IGP Support for affinity **exclude-any**.

Prerequisites for Flexible Algorithm

Segment routing must be enabled on the router before the Flexible Algorithm functionality is activated.

Restrictions for Flexible Algorithm

- A maximum of 20 IS-IS flexible algorithm sessions are supported.
- In IS-IS, the flexible algorithm affinity "exclude-any", "include-any", and "include-all" are supported.

Building Blocks of Segment Routing Flexible Algorithm

This section describes the building blocks that are required to support the SR Flexible Algorithm functionality in IS-IS and OSPF.

Flexible Algorithm Definition

Many possible constraints may be used to compute a path over a network. Some networks are deployed with multiple planes. A simple form of constraint may be to use a particular plane. A more sophisticated form of constraint can include some extended metric, like delay, as described in [RFC 8570]. Even more advanced case could be to restrict the path and avoid links with certain affinities. Combinations of these are also possible. To provide a maximum flexibility, the mapping between the algorithm value and its meaning can be defined by the user. When all the routers in the domain have the common understanding what the particular algorithm value represents, the computation for such algorithm is consistent and the traffic is not subject to looping. Here, since the meaning of the algorithm is not defined by any standard, but is defined by the user, it is called as Flexible Algorithm.

Flexible Algorithm Support Advertisement

An algorithm defines how the best path is computed by IGP. Routers advertise the support for the algorithm as a node capability. Prefix-SIDs are also advertised with an algorithm value and are tightly coupled with the algorithm itself.

An algorithm is a one octet value. Values from 128 to 255 are reserved for user defined values and are used for Flexible Algorithm representation.

Flexible Algorithm Definition Advertisement

To guarantee the loop free forwarding for paths computed for a particular Flexible Algorithm, all routers in the network must share the same definition of the Flexible Algorithm. This is achieved by dedicated router(s) advertising the definition of each Flexible Algorithm. Such advertisement is associated with the priority to make sure that all routers will agree on a single and consistent definition for each Flexible Algorithm.

Definition of Flexible Algorithm includes:

- Metric type
- Affinity constraints

To enable the router to advertise the definition for the particular Flexible Algorithm, **advertise-definition** command is used. At least one router in the area, preferably two for redundancy, must advertise the Flexible Algorithm definition. Without the valid definition being advertised, the Flexible Algorithm will not be functional.

Flexible Algorithm Prefix-SID Advertisement

To forward traffic on a Flexible Algorithm specific path, all routers participating in the Flexible Algorithm install an MPLS labeled path for the Flexible Algorithm specific prefix-SID. This Flexible Algorithm specific prefix-SID is advertised for the prefix. Only prefixes for which the Flexible Algorithm specific Prefix-SID is advertised, are subject to Flexible Algorithm specific forwarding.

Inter-Area Leaking

Effective Cisco IOS XE Bengaluru 17.4.1, Flexible Algorithm SIDs and prefixes are leaked between IS-IS areas. However, only the prefixes that are reachable by Level 1 or Level 2 paths are leaked. Similarly, only SIDs that are reachable in a given Flexible Algorithm are leaked.

For example, consider a prefix P:

- that is originated in Level 1 and leaked in to Level 2
- has SID value = 128 in Flexible Algorithm 128, and SID value = 129 in Flexible Algorithm 129
- for which Level 1 path exist only for SID value = 128, but not for SID value = 129

As a result of the above conditions, only SID 128 is leaked from Level 1 to Level 2 and not SID 129.

Calculation of Flexible Algorithm Path

A router may compute path for multiple Flexible Algorithms. A router must be configured to support particular Flexible Algorithm before it can compute any path for such Flexible Algorithm. A router must have a valid definition of the Flexible Algorithm before such Flexible Algorithm is used.

When computing the shortest path tree for particular Flexible Algorithm:

- All nodes that do not advertise support for such Flexible Algorithm will be pruned from the topology.
- If the Flexible Algorithm definition includes affinities that are excluded, then all links for which any of such affinities are advertised will be pruned from the topology.
- Router uses the metric that is part of the Flexible Algorithm definition. If the metric is not advertised for the particular link, such link will be pruned from the topology.

For OSPF and IS-IS, LoopFree Alternate (LFA) paths, and TI-LFA backup paths for a Flexible Algorithm are computed using the same constraints as the calculation of the primary paths for such Flexible Algorithm. These paths use Prefix-SIDs advertised specifically for such Flexible Algorithm to enforce a backup.

Installation of Forwarding Entries for Flexible Algorithm Paths

Flexible Algorithm paths to any prefix must be installed in the forwarding entries using the Prefix-SID that was advertised for such Flexible Algorithm. If the Prefix-SID for Flexible Algorithm is not known, such Flexible Algorithm path is not installed in forwarding for such prefix.

Only MPLS to MPLS entries are installed for a Flexible Algorithm path. No IP to IP or IP to MPLS entries are installed. These follow the native IPG paths computed based on the default algorithm and regular IGP metrics.

You can selectively filter the paths that are installed to the MFI by using the configuration command **distribute-list** *filter name* **in**. See Configuring Selective Path Filtering for configuration example. This feature is only supported for IS-IS Flexible Algorithm.

Flexible Algorithm Prefix-SID Redistribution

Prior to Cisco IOS XE 17.8, when prefixes were redistributed between protocols, only Prefix SIDs for SR algorithm 0 (regular SPF) were available.

In Cisco IOS XE 17.8, support for providing prefix SIDs for all supported algorithms when a prefix is redistributed is added. This feature is called the Segment Routing Flexible Algorithm Prefix SID Redistribution. This feature is enabled automatically when you configure redistribution of routes with strict or Flexible Algorithm SIDs.

When OSPF redistributes to ISIS, it redistributes all the algorithm prefixes and ISIS processes it. When ISIS redistributes to OSPF, only the base algorithm prefixes are processed by OSPF. Redistribution of other flexible algorithm prefixes are not supported in OSPF. For example, OSPF 10 is redistributed into ISIS 30, the strict SIDs and flexible algorithm SIDs are processed by ISIS. However, if ISIS 30 redistributes into OSPF 10, only the strict SIDs are processed by OSPF.

OSPF supports strict SPF and flexible algorithm. However, it does not support redistribution. For example, OSPF 10 and OSPF 20 are two instances having strict SPF and flexible algorithm. If OSPF 10 is redistributed into OSPF 20, then OSPF 20 will not process the strict SID and flexible algorithm SIDs of OSPF 10.

Displaying the Algorithm Information

You can use the **show mpls forwarding-table** command to display the non-zero algorithm specific prefix SID Label MPLS forwarding information. The command syntax is as follows:

show mpls forwarding <*ip*> <*mask*> [**algo** <*algo-number*>]

For more information, see Verifying the Flexible Algorithm Configuration, on page 327.

Flexible Algorithm Prefix Metric Advertisement

Segment Routing Flexible Algorithm Prefix Metric allows operators to associate metric computed inthe given Flexible Algorithm with a prefix during prefix inter-level leaking or inter-domain redistribution. It helps to compute the optimal inter-level or inter-domain path. When you configure Flexible Algorithm to support prefix-metric, the prefix-metric flag (M-flag) is advertised in ISIS Flexible Algorithm definition flags Sub-TLV. The sub-TLV is advertised only by the Level 1 and Level 2 routers. To view the prefix-metric flag (M-flag), use the **show isis database verbose** command. For more information, see Verifying the Flexible Algorithm Configuration, on page 327.

If a given flex algo algorithm (128-255) specifies the use of the Flex Algo Prefix Metric (FAPM), then the metric associated with the prefix *must* be advertised using the algorithm specific FAPM sub-TLV by the ABRs which advertise the prefix into other levels/areas. When the Flexible Algorithm Definition specifies the use of FAPM (M-flag) then only prefixes which have an algorithm specific FAPM advertisement will be considered reachable in the algorithm specific topology.

Note Cisco IOS XE supports flexible algorithm prefix-metric insertion only during prefix inter-level leaking and not during inter-domain redistribution.

The ISIS Flexible Algorithm Prefix Metric Sub-TLV supports the advertisement of a Flexible Algorithm specific prefix metric associated with a given prefix advertisement.

The following command is used to enable advertisement of the Flexible Algorithm Prefix Metric:

```
router isis 1
flex-algo 128
  advertise-definition
  prefix-metric

# show isis 1 rib redistribution level-2
IPv4 redistribution RIB for IS-IS process 1
IPV4 unicast base topology (TID 0, TOPOID 0x0) ==================
====== Level 2 ======
10.1.1.1/32
  [ISIS/20] isis prefix-SID index: 1, R:1 N:1 P:1 E:0 V:0 L:0
  flex-algo 128 SID index: 11, R:0 N:1 P:0 E:0 V:0 L:0 map 0x0
    prefix-metric: 20, advertised
```

In this command output, you can see that the prefix-metric is advertised only if it is enabled.

Flexible Algorithm Configurations

This section describes various configurations that are required to support the SR Flexible Algorithm functionality.

Table 44: Flexible Algorithm Configuration

Task	Protocol	Mode	Command
Configure Flexible Algorithm	IS-IS and OSPF	IS-IS and OSPFconfiguration sub-mode	**flex-algo** *algorithm number* *algorithm number* —value from 128 to 255
Setting metric type	IS-IS and OSPF	Flex-algo sub-mode	**[IS-IS]** **metric type {delay | te}** **Note** By default, the regular IGP metric is used. If the delay metric is enabled, the advertised delay on the link is used as the metric for flexible algorithm computation. If TE metric is enabled, the advertised TE metric in the link is used as the metric for flexible algorithm computation. **[OSPF]** **metric-type {delay | te-metric | igp-metric}**

Task	Protocol	Mode	Command
Setting affinity	IS-IS and OSPF	Flex-algo sub-mode	**[IS-IS]** **affinity {exclude-any | include-any | include-all} name** *affinity-name* **[OSPF]** **affinity {exclude-any | include-any | include-all} name** *affinity-name* *affinity-name*: Name of the affinity map
Setting priority	IS-IS and OSPF	Flex-algo sub-mode	**[IS-IS and OSPF]** **priority** *priority value* *priority-value*: Priority used during the Flexible Algorithm definition election
Enable advertisement of the Flexible Algorithm definition in IS-IS and OSPF: Affinity-map associates the name with the particular bit positions in the Extended Admin Group bitmask	IS-IS and OSPF		**[IS-IS]** **affinity-map** *affinity-name* **bit-position***bit number* **[OSPF]** **affinity-map name** *affinity-name* **bit-position***bit number* *affinity-name*: Name of the affinity map *bit number*: Bit position in the Extended Admin Group bitmask

Task	Protocol	Mode	Command
Associate affinity with an interface	IS-IS and OSPF		**[IS-IS]** **isis affinity flex-algo name** *affinity-name* **[OSPF]** **ip ospf affinity flex-algo name** *affinity-name* *affinity-name*: Name of the affinity map

From Cisco IOS XE Release 17.11.1a, a new metric, **TE**, is introduced for IS-IS Flexible Algorithm. This metric includes a new keyword for the **isis flex-algo metric-type** command.

isis *instance* **flex-algo** *algo* **metric-type {delay | te}**

This keyword is available for Cisco ASR 1000 series platforms.

Note By default, the IGP metric is used for flexible algorithm computation. If either the delay or the TE metric is enabled, the advertised delay or the TE metric in the link is used as the metric for flexible algorithm computation.

Command for Prefix SID in Flexible Algorithm Configuration

To define a prefix SID associated with a specific flexible algorithm, a new command is added under segment routing, for both connected prefix SID map, as well as the mapping server:

```
segment-routing mpls
connected-prefix-sid-map
address-family ipv4algorithmflex-algo
ip addressmask [index | absolute] sid range range of SIDs
```

```
segment-routing mpls
mapping-server
prefix-sid-map
address-family ipv4algorithmflex-algo
ip addressmask [index | absolute] sid range range of SIDs
```

Configuring IS-IS Flexible Algorithm

The following is an example of how to configure the IS-IS flexible algorithm:

```
router isis 1
 net 49.0002.0000.0001.00
 is-type level-1
 metric-style wide
 log-adjacency-changes
 nsf cisco
 distribute link-state
 segment-routing mpls
 segment-routing prefix-sid-map advertise-local
```

```
 affinity-map blue bit-position 8
 affinity-map green bit-position 201
 affinity-map red bit-position 65

 fast-reroute per-prefix level-1 all
 fast-reroute tie-break level-1 node-protecting 100
 fast-reroute tie-break level-1 srlg-disjoint 50
 fast-reroute ti-lfa level-1
 fast-reroute ti-lfa level-2
 microloop avoidance segment-routing
 microloop avoidance rib-update-delay 10000

flex-algo 129
  advertise-definition
  metric-type delay
  priority 120
  affinity
   exclude-any
    name red
   !
```



Note Use the **fast-reroute disable** command to disable TI LFA.

The following example shows how to configure the IS-IS flexible algorithm with metric-type as TE:

```
router isis 1
net 49.0002.0000.0001.00
is-type level-1
metric-style wide
log-adjacency changes
nsf cisco
distribute link-state
segment-routing mpls
segment-routing prefix-sid-map advertise-local
affinity-map blue bit-position 8
affinity-map green bit-position 201
affinity-map red bit-position 65

fast-reroute per-prefix level-1 all
fast-reroute tie-break level-1 node-protecting 100
fast-reroute tie-break level-1 srlg-disjoint 50
fast-reroute ti-lfa level-1
fast-reroute ti-lfa level-2
microloop avoidance segment-routing
microloop avoidance rib-update-delay 10000

flex-algo 129 advertise-definition
metric-type te
  priority 120
affinity exclude-any name red
!
```

The following example shows how to configure IS-IS TE metric on an interface:

```
interface Ethernet0/0
ip address 10.12.12.1 255.255.255.0
ip router isis 1
ipv6 address 2001:20::1/112
ipv6 router isis 1
isis network point-to-point
isis te-metric flex-algo 500
```

Redistributing IS-IS

The following example shows how to redistribute IS-IS:

```
router isis 2
router-id Loopback0
metric-style wide
segment-routing mpls
segment-routing prefix-sid-map advertise-local
flex-algo 128
advertise-definition
redistribute isis 1 ip level2 <-----
passive-interface Loopback0
mpls traffic-eng level-1
mpls traffic-eng level-2
```

Configuring SRTE-ODN Association

The following example shows how to configure an SR traffic engineering - ODN association:

```
segment-routing traffic-eng
 on-demand color 100
  authorize
  candidate-paths
   preference 100
    constraints
     segments
      dataplane mpls
      algorithm 129
     !
    !
    dynamic
     metric
      type delay
    !
   !
```

Configuring the Interface for Flexible Algorithm

The following example shows how to configure an interface for flexible algorithm:

```
interface GigabitEthernet0/0/6
 ip address 10.11.11.1 255.255.255.0
 ip router isis 1
 mpls ip
 mpls traffic-eng tunnels
 bfd template pw_bfd
 isis network point-to-point
 isis affinity flex-algo
  name red
  !
```

Configuring BGP

The following example shows how to configure BGP:

```
router bgp 100
 bgp router-id 10.1.1.1
```

```
 bgp log-neighbor-changes
 bgp graceful-restart
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 ha-mode sso
 neighbor 10.2.2.2 update-source Loopback1
 !
 address-family ipv4
  neighbor 10.2.2.2 activate
 exit-address-family
 !
address-family vpnv4
  neighbor 10.2.2.2 activate
  neighbor 10.2.2.2 send-community both
  neighbor 10.2.2.2 route-map BGP_TE_MAP out
 exit-address-family
 !
 address-family ipv4 vrf SR
  redistribute connected
  neighbor 10.132.1.1 remote-as 101
  neighbor 10.132.1.1 activate
 exit-address-family
!
```

Configuring Selective Path Filtering

The following example shows how to selectively filter the paths that are installed in the MPLS Forwarding Infrastructure (MFI):

```
Prefix-source
=====================
interface Loopback1
ip address 10.1.1.1 255.255.255.255
ip router isis
 isis tag 111

Remote router configured for selective path filtering
 ================================================
route-map block deny 10
match tag 111
!
route-map block permit 100
!
router isis 1
!
flex-algo 135
!
distribute-list route-map block in
```

Configuring SR Policy with PCE Delegation

The following example shows how to configure SR policy with Path Computation Element (PCE) delegation:

```
policy p-delay
  color 1111 end-point 10.6.6.6
  candidate-paths
   preference 1
    constraints
     segments
      dataplane mpls
```

```
  algorithm 128
 !
!
dynamic
 pcep
```

Verifying the Flexible Algorithm Configuration

The following is a sample output of the **show isis flex-algo** *value* command showing all the information regarding the IS-IS flexible algorithm:

```
show isis flex-algo 129
Tag 1:
IS-IS Flex-Algo Database
 Flex-Algo count: 7

Flex-Algo 129:
    IS-IS Level-1
      Definition Priority: 222
      Definition Source: R2-RSP3-2015.00, (Local)
      Definition Equal to Local: Yes
      Definition Metric Type: Delay
      Definition Flex-Algo Prefix Metric: No
      Disabled: No
      Microloop Avoidance Timer Running: No
   Local Priority: 222
   FRR Disabled: No
   Microloop Avoidance Disabled: No
```

The following is a sample output for the **show isis flex-algo** command showing the metric type TE:

```
show isis flex-algo 129 Tag 1:
IS-IS Flex-Algo Database Flex-Algo count: 7

Flex-Algo 129:
IS-IS Level-1
Definition Priority: 222
Definition Source: R2-RSP3-2015.00, (Local) Definition Equal to Local: Yes
Definition Metric Type: TE
Definition Flex-Algo Prefix Metric: No Disabled: No
Microloop Avoidance Timer Running: No Local Priority: 222
FRR Disabled: No
Microloop Avoidance Disabled: No
```

The following is a sample output of the **show isis rib flex-algo** *value* command showing all the IS-IS local RIB information:

```
show isis rib flex-algo 129
IPv4 local RIB for IS-IS process 1

IPV4 unicast topology base (TID 0, TOPOID 0x0) ================= Repair path attributes:
DS - Downstream, LC - Linecard-Disjoint, NP - Node-Protecting PP - Primary-Path, SR -
SRLG-Disjoint

Flex-algo 129

10.1.1.1/32 prefix attr X:0 R:0 N:1 source router id: 10.1.1.1 SID index 38 - Bound
[115/L1/113] via 10.11.11.1(GigabitEthernet0/4/6) R1-ASR920-2011.00-00, from 10.1.1.1, tag
 0
LSP 6/6/351(351), prefix attr: X:0 R:0 N:1 Source router id: 10.1.1.1
Prefix-SID index: 38, R:0 N:1 P:0 E:0 V:0 L:0
```

```
label: implicit-null
repair path: 10.20.20.2 (GigabitEthernet0/4/7) metric: 117 (DS,SR) local LFA
label: implicit-null
repair source: R1-ASR920-2011, LSP 6

10.2.2.2/32 prefix attr X:0 R:0 N:1 source router id: 10.2.2.2 SID index 39 - Bound
[115/L1/24] via 10.13.13.2(GigabitEthernet0/1/5) R4-RSP3-2036.00-00, from 10.2.2.2, tag 0
LSP 2/3/345(345), prefix attr: X:0 R:0 N:1 Source router id: 10.2.2.2
Prefix-SID index: 39, R:0 N:1 P:0 E:0 V:0 L:0
label: 17039
repair path: 10.4.4.4 (MPLS-SR-Tunnel4) metric: 170 (DS,NP,SR) next-hop: 10.20.20.2
(GigabitEthernet0/4/7)
TI-LFA node/SRLG-protecting, SRLG-protecting
SRGB: 17000, range: 7000 prefix-SID index: 39, R:0 N:1 P:0 E:0 V:0 L:0
label: 17039
P node: R3-RSP2-2013[10.4.4.4], label: 17221
repair source: R6-RSP3-2038, LSP 3

10.4.4.4/32 prefix attr X:0 R:0 N:1 source router id: 10.4.4.4 SID index 221 - Bound



[115/L1/172] via 10.13.13.2(GigabitEthernet0/1/5) R4-RSP3-2036.00-00, from 10.4.4.4, tag
0
LSP 2/7/24(24), prefix attr: X:0 R:0 N:1 Source router id: 10.4.4.4
Prefix-SID index: 221, R:0 N:1 P:0 E:0 V:0 L:0
label: 17221
repair path: 10.20.20.2 (GigabitEthernet0/4/7) metric: 184 (DS,NP,SR) local LFA
label: 17221
repair source: R3-RSP2-2013, LSP 7

10.5.5.5/32 prefix attr X:0 R:0 N:1 source router id: 10.5.5.5 SID index 222 - Bound
[115/L1/17] via 10.13.13.2(GigabitEthernet0/1/5) R4-RSP3-2036.00-00, from 10.5.5.5, tag 0
LSP 2/2/347(347), prefix attr: X:0 R:0 N:1 Source router id: 10.5.5.5
Prefix-SID index: 222, R:0 N:1 P:0 E:0 V:0 L:0
label: implicit-null
repair path: 10.4.4.4 (MPLS-SR-Tunnel4) metric: 170 (DS,SR) next-hop: 10.20.20.2
(GigabitEthernet0/4/7)
TI-LFA SRLG-protecting
SRGB: 17000, range: 7000 prefix-SID index: 222, R:0 N:1 P:0 E:0 V:0 L:0
label: 17222
P node: R3-RSP2-2013[10.4.4.4], label: 17221
repair source: R4-RSP3-2036, LSP 2

10.6.6.6/32 prefix attr X:0 R:0 N:1 source router id: 10.6.6.6 SID index 333 - Bound
[115/L1/122] via 10.13.13.2(GigabitEthernet0/1/5) R4-RSP3-2036.00-00, from 10.6.6.6, tag
0
LSP 2/4/351(351), prefix attr: X:0 R:0 N:1 Source router id: 10.6.6.6
Prefix-SID index: 333, R:0 N:1 P:0 E:0 V:0 L:0
label: 17333
repair path: 10.4.4.4 (MPLS-SR-Tunnel4) metric: 170 (DS,NP,SR) next-hop: 10.20.20.2
(GigabitEthernet0/4/7)
TI-LFA node/SRLG-protecting, SRLG-protecting
SRGB: 17000, range: 7000 prefix-SID index: 333, R:0 N:1 P:0 E:0 V:0 L:0
label: 17333
P node: R3-RSP2-2013[10.4.4.4], label: 17221
repair source: R5-ASR920-2012, LSP 4
```

The following is a sample output of the **show isis topo flex-algo** *value* command showing information regarding the IS-IS paths to intermediate systems:

```
show isis topo flex-algo 129
Tag 1:
IS-IS TID 0 paths to level-1 routers
```

```
Flex-algo 129
System Id            Metric     Next-Hop            Interface   SNPA
920_1                3          RSP2_2              Gi0/15/0    e8ed.f3b8.f804
RSP3_R1              **
RSP2_1               2          RSP2_2              Gi0/15/0    e8ed.f3b8.f804
RSP3_R2              **
RSP2_2               1          RSP2_2              Gi0/15/0    e8ed.f3b8.f804
RSP3_R3              --
```

The following is a sample output of the **show isis fast-reroute ti-lfa tunnel** command showing information regarding the IS-IS TI-LFA tunnels:

```
show isis fast-reroute ti-lfa tunnel
Tag null:
Fast-Reroute TI-LFA Tunnels:
Tunnel  Interface  Next Hop         End Point        Label     End Point Host
Tag 1:
Fast-Reroute TI-LFA Tunnels:

Tunnel  Interface  Next Hop         End Point        Label     End Point Host

MP2     Gi0/0/6    10.12.12.2       10.2.2.2          17019     RSP3_R3

MP5     Gi0/0/5    10.11.11.2       10.2.2.2          17019     RSP3_R3

MP3     Gi0/0/6    10.12.12.2       10.6.6.6          17333     RSP2_2

                                    10.2.2.2          16        RSP3_R3

MP9     Gi0/0/5    10.11.11.2       10.2.2.2          17039     RSP3_R3

MP1     Gi0/0/6    10.12.12.2       10.6.6.6          20333     RSP2_2

                                    10.2.2.2          16        RSP3_R3

MP6     Gi0/0/5    10.11.11.2       10.2.2.2          17049     RSP3_R3
```

The following is a sample output of the **show ip ospf topology** command showing the node and link information compiled from the link-state advertisements (LSAs):

```
R1#show ip ospf topology
        Process OSPF-10

  Instance : global
   Router ID : 10.1.1.1
    Area :  (8 nodes)
      Node : 10.2.0.2 (pseudo) (2 links)
        Link : 10.1.1.1 10.0.0.0 Transit
        Link : 10.1.1.2 10.0.0.0 Transit
      Node : 10.1.1.1 (root) (3 links) ABR
        Algos supported: 128, 129
        Flex Algo Definition: 128
        Flex Algo Definition: 129
        Link : 10.1.1.6 10.0.0.2 Point-to-point
        Link : 10.1.1.6 10.6.1.1 Point-to-point
        Link : 10.2.0.2 10.2.0.1 Transit
      Node : 10.1.1.2  (3 links)
        Algos supported: 128
        Link : 10.1.1.3 10.3.0.2 Point-to-point
        Link : 10.1.1.54 10.5.0.2 Point-to-point
        Link : 10.2.0.2 10.2.0.2 Transit
      Node : 10.1.1.3  (2 links)
        Algos supported: 128
        Link : 10.1.1.2 10.3.0.3 Point-to-point
```

```
      Link : 10.1.1.4 10.4.0.3 Point-to-point
    Node : 10.1.1.4  (3 links) ABR, ASBR
      Algos supported: 128, 129
      Link : 10.1.1.3 10.4.0.4 Point-to-point
      Link : 10.1.1.9 10.0.0.3 Point-to-point
      Link : 10.1.1.54 10.5.0.4 Point-to-point
    Node : 10.1.1.6  (4 links)
      Algos supported: 129
      Link : 10.1.1.1 10.0.0.2 Point-to-point
      Link : 10.1.1.1 10.6.1.6 Point-to-point
      Link : 10.1.1.54 10.6.0.6 Point-to-point
      Link : 10.1.1.54 10.6.1.6 Point-to-point
    Node : 10.1.1.9  (1 links) ABR
      Link : 10.1.1.4 10.0.0.3 Point-to-point
    Node : 10.1.1.54  (4 links)
      Algos supported: 129
      Link : 10.1.1.2 10.5.0.5 Point-to-point
      Link : 10.1.1.4 10.5.0.5 Point-to-point
      Link : 10.1.1.6 10.6.0.5 Point-to-point
      Link : 10.1.1.6 10.6.1.5 Point-to-point
  Area :  (2 nodes)
    Node : 10.1.1.1 (root) (1 links) ABR
      Algos supported: 128, 129
      Flex Algo Definition: 128
      Flex Algo Definition: 129
      Link : 10.1.1.8 10.8.0.1 Point-to-point
    Node : 10.1.1.8  (1 links)  ASBR
      Link : 10.1.1.1 10.8.0.8 Point-to-point
```

The following is a sample output of the **show ip ospf topology prefix** command showing the node and prefix information compiled from the LSAs:

```
R1#show ip ospf topology prefix
        Process OSPF-10

  Instance : global
   Router ID : 10.1.1.1
    Area :  (8 nodes)
      Node : 10.2.0.2 (pseudo) (2 links)
      Node : 10.1.1.1 (root) (3 links) ABR
        Algos supported: 128, 129
        Flex Algo Definition: 128
        Flex Algo Definition: 129
      Node : 10.1.1.2  (3 links)
        Algos supported: 128
      Node : 10.1.1.3  (2 links)
        Algos supported: 128
        Prefix : 10.1.1.34/32
      Node : 10.1.1.4  (3 links) ABR, ASBR
        Algos supported: 128, 129
        Prefix : 10.1.1.4/32
        Prefix : 10.1.1.34/32
        Prefix : 10.1.1.45/32
      Node : 10.1.1.6  (4 links)
        Algos supported: 129
      Node : 10.1.1.9  (1 links) ABR
      Node : 10.1.1.54  (4 links)
        Algos supported: 129
        Prefix : 10.1.1.54/32
    Area :  (2 nodes)
      Node : 10.1.1.1 (root) (1 links) ABR
        Algos supported: 128, 129
        Flex Algo Definition: 128
        Flex Algo Definition: 129
      Node : 10.1.1.8  (1 links)  ASBR
```

The following is a sample output of the **show ip ospf topology route** command showing the path information of routes computed based on route calculation:

```
R1#show ip ospf topology route
 Route Table of OSPF-10 with router ID 10.1.1.1 (VRF global)

  10.1.1.4/32
    Algo 128, Metric 31, SID 132, Label 16132
      10.2.0.2, from 10.1.1.2, via Ethernet0/1
    Algo 129, Metric 31, SID 133, Label 16133
      10.1.1.6, from 10.1.1.6, via Ethernet0/0
      10.6.1.6, from 10.1.1.6, via Ethernet0/3
  10.1.1.34/32
    Algo 128, Metric 21, SID 43, Label 16043
      10.2.0.2, from 10.1.1.2, via Ethernet0/1
  10.1.1.45/32
    Algo 129, Metric 31, SID 4294967295, Label 1048577
      10.1.1.6, from 10.1.1.6, via Ethernet0/0
      10.6.1.6, from 10.1.1.6, via Ethernet0/3
  10.1.1.54/32
    Algo 129, Metric 21, SID 45, Label 16045
      10.1.1.6, from 10.1.1.6, via Ethernet0/0
      10.6.1.6, from 10.1.1.6, via Ethernet0/3
```

The following is a sample output of the **show mpls forwarding-table** command showing the non-zero algorithm specific prefix SID Label MPLS forwarding information:

```
#show mpls forwarding-table 10.23.23.23 255.255.255.255 algo 20
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
18         16023      0-10.23.23.23/32-4 (10:30:20:1)   \
                                       0             Et1/1      10.1.1.2
```

The prefix or tunnel ID column provides information about the metric, for example, 0-10.6.6.6/32-4 (4:50:128:0).

The four parts next to the prefix are as follows:

- pdb-index=4
- metric=50
- algo=128
- via-srms=0

The **via-srm**s field indicates whether the source of the label came from a prefix reachability advertisement (0) or from a mapping server advertisement (1). Labels derived from mapping server advertisements should not be advertised when a redistributed route is advertised by the destination protocol for redistribution.

The **pdb-index** field indicates the protocol instance. The following command output shows the different protocols and their values:

```
# show ip protocols summary
Index Process Name
0 connected
1 static
2 application
3 nat-route
4 isis 1
```

The following is a sample output of the **show isis rib redistribution** command showing the redistributed prefix:

```
# show isis rib redistribution

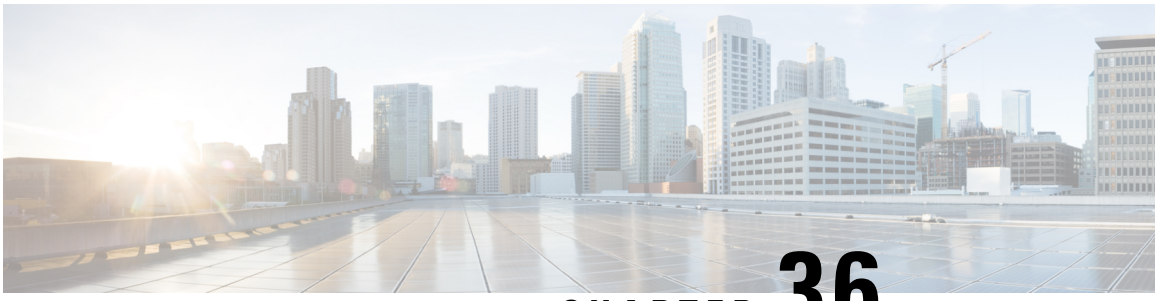
IPv4 redistribution RIB for IS-IS process 1

IPV4 unicast base topology (TID 0, TOPOID 0x0) =================
====== Level 1 ======
====== Level 2 ======
10.3.3.3/32
  [Connected/0]  prefix-SID index: 31, R:0 N:1 P:0 E:0 V:0 L:0
  strict-SPF SID index: 32, R:0 N:1 P:0 E:0 V:0 L:0
  flex-algo 128 SID index: 33, R:0 N:1 P:0 E:0 V:0 L:0 map 0x1
    prefix-metric: 0, not advertised
10.4.4.4/32
  [ISIS/0] external interarea  prefix-SID index: 41, R:1 N:0 P:1 E:0 V:0 L:0
  strict-SPF SID index: 42, R:1 N:0 P:1 E:0 V:0 L:0
  flex-algo 128 SID index: 43, R:1 N:0 P:1 E:0 V:0 L:0 map 0x0
    prefix-metric: 40, not advertised
  prefix attr: X:1 R:0 N:0
```

In this example, you can see the strict SID or flexible algorithm prefix SIDs. The redistributed prefix is noted as the inter area route, and the X flag is set.

The following is a sample output of the **show isis database verbose** command showing the prefix-metric flag (M-flag) that is advertised in ISIS flexible algorithm definition flags sub-TLV:

```
# show isis database verbose
..
Router CAP:   10.1.1.1, D:0, S:0
    Segment Routing: I:1 V:0, SRGB Base: 16000 Range: 8000
    Segment Routing Local Block: SRLB Base: 15000 Range: 1000
    Node-MSD
      MSD: 16
    Flex algorithm: 150 Metric-Type: IGP Alg-type: SPF Priority: 128
    Segment Routing Algorithms: SPF, Strict-SPF, Flex-algo 128
    Segment Routing Algorithms: Flex-algo 150
    Flex algorithm: 128 Metric-Type: IGP Alg-type: SPF Priority: 128
      Flex-Algo Definition Flags:
       M:1.
```



CHAPTER 36

L2VPN over SR-TE Preferred Path

Table 45: Feature History

Feature Name	Release Information	Description
L2VPN Traffic Steering Using SR-TE Preferred Path with Flexible Algorithm	Cisco IOS XE Bengaluru 17.6.1	This feature allows you to configure an SR policy with as the preferred path for a VPWS or VPLS pseudowire, with Flexible Algorithm. VPWS or VPLS pseudowires between same PEs can be routed over different SR policies based on the requirements. Prior to this release, you could only steer the traffic using the SR policy for routing IPv4 traffic to a destination pseudowire (over IGP or BGP-LU).

Virtual Private LAN Services (VPLS) enables enterprises to link together multiple Ethernet-based LANs via the infrastructure provided by their service provider.

VPLS uses the service provider core to join multiple attachment circuits of an enterprise to simulate a virtual bridge. From the enterprise point of view, there is no topology for VPLS. All customer edge (CE) devices appear to connect to a logical bridge emulated by the service provider core.

Prior to Cisco IOS XE Bengaluru Release 17.6.1, L2VPN (VPLS or VPWS) traffic over SR policies could not be steered. You could only steer IPv4 traffic using the SR policy for routing IPv4 traffic to a destination pseudowire (over IGP or BGP-LU).

Now you configure an SR policy as the preferred path for a VPWS or VPLS pseudowire, with Flexible Algorithm. VPWS or VPLS pseudowires between same PEs can also be routed over different SR policies.

Disable Fallback Option

The disable fallback option disables the router from using the default path when the preferred path SR policy goes down.

- Restrictions, on page 334
- Configuring L2VPN Traffic Steering Using SR-TE Preferred Path with Flexible Algorithm, on page 334
- Configuration Example 1: VPWS Psuedowire over SR-TE Preferred Path, on page 336
- Configuration Example 2: VPWS Psuedowire over SR-TE Preferred Path, on page 336

• Configuration Example 3: VPLS Psuedowire over SR-TE Preferred Path, on page 337
• Verification of L2VPN over SR-TE Preferred Path Configuration, on page 337

Restrictions

- You cannot add On-Demand (ODN) policies to the preferred path.
- L2VPN over SR-TE preferred path is only supported on SR Per Destination Policy (PDP); and not on the SR Per-Flow Policy (PFP).
- L2VPN over SR-TE preferred path can only be configured using the pseudowire interface.
- This feature is supported only on IS-IS protocol

Configuring L2VPN Traffic Steering Using SR-TE Preferred Path with Flexible Algorithm

To configure IS-IS with Flex Algo:

```
router isis 1

affinity-map green bit-position 0
affinity-map red bit-position 1
affinity-map yello bit-position 2
flex-algo 128
  advertise-definition
  metric-type delay
  priority 200
  affinity
   exclude-any
    name red
    name yellow
!
flex-algo 129
  advertise-definition
  priority 200
  affinity
   exclude-any
    name green
    name red



interface Tunnel100
isis affinity flex-algo
  name green
  !
interface Tunnel101
isis affinity flex-algo
  name yellow
 !
interface Tunnel102
isis affinity flex-algo
  name red
```

```
segment-routing traffic-eng
 policy p-2000
  color 2000 end-point 10.4.4.4
  performance-measurement
   delay-measurement
  candidate-paths
   preference 10
    constraints
     segments
      dataplane mpls
      algorithm 128
     !
     !
    !
    dynamic
```

To create SR static policy for MPLS label:

```
configure terminal segment-routing traffic-eng
segment-list name segment-name
index 1 mpls label first hop label
index 2 mpls label second hop label !
policy policy-name
color color-code end-point destination IP Address candidate-paths
preference preference
explicit segment-list segment-name
constraints
segments dataplane mpls
```

You can also create SR static policy for the following:

- MPLS adjacency
- MPLS prefix

L2VPN over SR-TE preferred path can be configured in the following ways:

- Non-Template based Configuration
- Template-based Configuration

Non-template Based Configuration:

- **Create Pseudowire**

  To create pseudowire:

  ```
  interface pseudowire 1
   encapsulation mpls
   neighbor peer-address vc-id
  ```

- **Attach Policy Using Preferred Path**

  To attach a policy using the preferred path:

  ```
  interface pseudowire1
   preferred-path segment-routing traffic-eng policy policy-name [disable-fallback]
  ```

Template-based Configuration:

- **Create Template Type Pseudowire**

  To create template type pseudowire:

```
template type pseudowire name
 encapsulation mpls
 preferred-path segment-routing traffic-eng policy name [disable-fallback]
```

- **Attach Policy Using Preferred Path**

  To attach a policy using the preferred path:

```
interface pseudowire 1
 source template type pseudowire name
```

Configuration Example 1: VPWS Psuedowire over SR-TE Preferred Path

```
!
interface
gi0/0/1
service instance 1000
ethernet encapsulation
dot1q 1000 !
template type pseudowire l2vpntest
 encapsulation mpls
preferred-path Segment-Routing traffic-eng policy p106
l2vpn xconnect context l2vpn-test
member 10.6.6.6 1000 template
l2vpntest member gi0/0/1
service-instance 1000  !
```

Configuration Example 2: VPWS Psuedowire over SR-TE Preferred Path

```
!
!
interface gi0/0/1
service instance 1000 ethernet
 encapsulation dot1q 1000
 !
template type pseudowire
l2vpntest encapsulation mpls
preferred-path Segment-Routing traffic-eng policy p106 !

interface pseudowire1000
 source template type pseudowire l2vpntest
encapsulation mpls  neighbor 10.1.1.1 1000 !

l2vpn xconnect context l2vpn-test
member pseudowire 1000
 member gi0/0/1 service-instance 1000
```

Configuration Example 3: VPLS Psuedowire over SR-TE Preferred Path

```
interface gi0/0/1

service instance 1000
ethernet   encapsulation
dot1q 1000 !

interface pseudowire106
encapsulation mpls
 neighbor 10.6.6.6 1000

 preferred-path Segment-Routing traffic-eng policy p106 !
interface pseudowire104
 encapsulation mpls
  neighbor 10.4.4.4 1000

preferred-path Segment-Routing traffic-eng policy p104
 !
l2vpn vfi context VC_1000   vpn id 1000  member
 pseudowire106  member pseudowire104
!

bridge-domain 1000

member gi0/0/1 service-instance
 1000 member vfi VC_1000
```

Verification of L2VPN over SR-TE Preferred Path Configuration

Use the **show segment-routing traffic-eng policy name** *policy name* **detail** command to verify the policy configuration:

```
Router#show segment-routing traffic-eng policy name CE11-PE12 detail

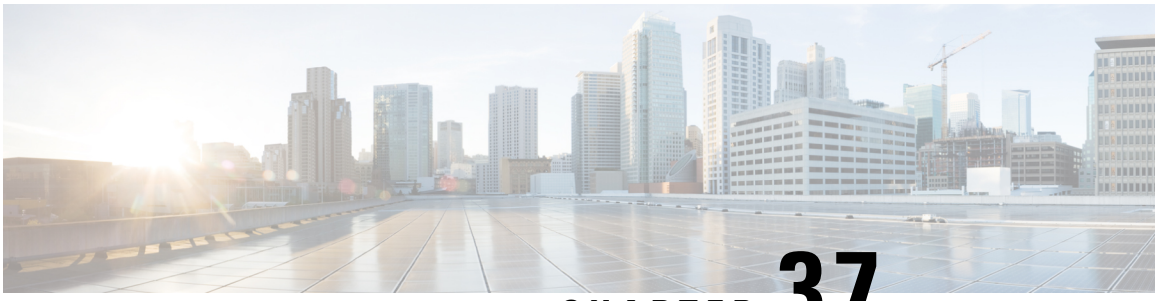
Name: CE11-PE12 (Color: 50 End-point: 10.12.12.12)
  Owners : CLI
  Status:
    Admin: up, Operational: up for 70:04:00 (since 08-17 07:55:36.536)
  Candidate-paths:
    Preference 100 (CLI):
      Explicit: segment-list IntraDomain (active)
        Weight: 1, Metric Type: TE
16005
16008
16010
Attributes:
    Binding SID: 20
      Allocation mode: dynamic
      State: Programmed
  Tunnel ID: 65538 (Interface Handle: 0x20)
  Per owner configs:
    CLI
      Binding SID: dynamic
  Stats:
    Packets: 0  Bytes: 0
```

```
  Event history:
    Timestamp                  Client                  Event type              Context:
Value
    ----------------           ------                  --------------
---------: ---------
    10-28 04:05:37.028         L2VPN                   Policy created          Name: L2VPN

    10-28 04:05:37.048         L2VPN                   BSID allocated          FWD: label
 20
    10-28 04:05:37.494         L2VPN                   Client removed          Owner:
Destroyed
    10-28 04:05:37.494         CLI                     Set colour                Colour:
 230
    10-28 04:05:37.494         CLI                     Set end point           End-point:
 12.12.12.12
    10-28 04:05:37.496         CLI                     Set explicit path       Path option:
 IntraDomain
    10-28 04:08:22.873         FH Resolution           Policy state UP         Status:
PATH RESOLVED
    10-28 04:08:45.630         FH Resolution           REOPT triggered         Status:
REOPTIMIZED
```

Use **show mpls l2transport vc 1000 detail** command to verify the L2VPN over SR-TE preferred path:

```
Router#show mpls l2transport vc 1000 detail
Local interface: VFI VC_1000 vfi up
  Interworking type is Ethernet
  Destination address: 10.12.12.12, VC ID: 1000, VC status: up
    Output interface: tu65538, imposed label stack {16005 16008 16010 32}
    Preferred path: not configured
    Default path: active
    Next hop: 10.168.1.1
  Create time: 1w4d, last status change time: 22:50:57
    Last label FSM state change time: 22:51:46
  Signaling protocol: LDP, peer 10.1.1.1:0 up
    Targeted Hello: 10.2.2.2(LDP Id) -> 10.1.1.1, LDP is UP
    Graceful restart: not configured and not enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 26, remote 21
    Group ID: local n/a, remote 16
    MTU: local 9000, remote 9000
    Remote interface description:
    MAC Withdraw: sent:0, received:301
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense
```



CHAPTER 37

COE-PCE Initiated SR Policy with IGP Autoroute Announce

Table 46: Feature History

Feature Name	Release Information	Feature Description
PCE Initiated SR Policy with IGP Autoroute Announce	Cisco IOS XE Bengaluru 17.7.1a Cupertino	This feature enables a steering mechanism in which IGPs automatically use the policy for destination's downstream of the policy end point.

As part of a tactical TE solution, the Path Computation Element (PCE) can provision a Segment Routing Traffic Engineering (SR-TE) policy to mitigate link congestion.

Autoroute announcement is a steering mechanism in which IGPs automatically use the policy for destination's downstream of the policy end point. Autoroute announcement is performed using Cisco Crossworks Optimization Engine (COE). COE provides real-time network optimization allowing operators to maximize network utilization effectively and increase service velocity.

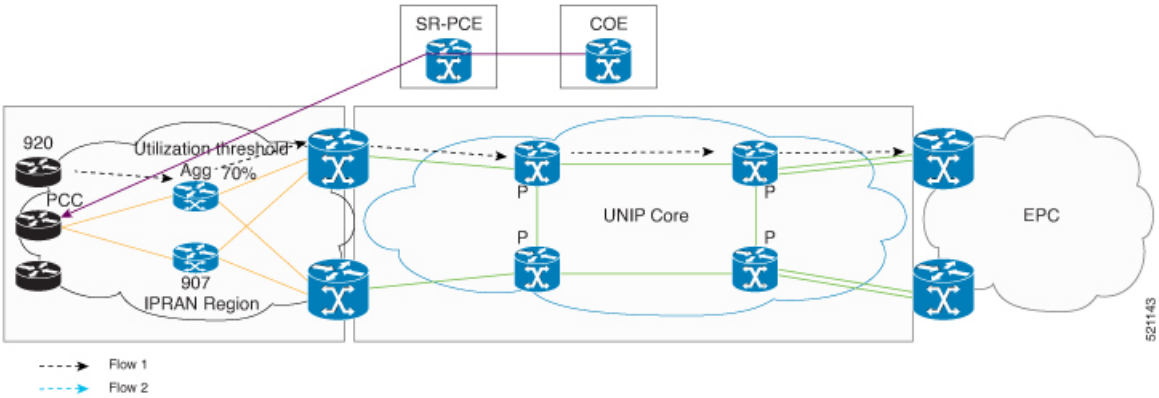
A PCE collects various pieces of network information to determine traffic flows causing link congestion. The PCE computes a suitable path to divert those flows and to alleviate the congestion. The PCE then deploys the SR-TE policy to divert the traffic leading to the congestion using the Stateful Path Computation Element Protocol (PCEP) to provision the policy. When the congestion is alleviated, the SR-TE policy is removed.

The PCEP message contains SID list to be deployed by the head-end. Path Computation Client (PCC) profiles allow activation of autoroute announce for the policy provisioned by PCEP, using the profile IDs. The profile ID on the PCE and PCC should match, otherwise the policy is not provisioned. For example, if the PCE provisions a policy with profile ID 1 and the head-end where the policy is being provisioned also has the PCC profile ID 1 configured with autoroute announce, COE-PCE initiated SR policy is activated for that policy.

- COE-PCE Initiated SR Policy, on page 340
- ECMP Over SR-TE, on page 341

COE-PCE Initiated SR Policy

Figure 38: COE-PCE Initiated SR Policy



The preceding topology shows how an SR-PCE policy is initiated from COE:

- SR policy is configured on the COE with profile ID.
- COE pushes the SR policy to PCE and PCE forwards the SR policy to PCC.
- Profile ID on PCC is matched with the profile ID on COE-PCE.
- IGP autoroute announce is configured on the PCC.
- The policy gets provisioned.
- The data traffic now adheres to the SR policy that is pushed from the COE.
- Complete SR Policy manipulation occurs only on COE.

Restrictions for PCE Initiated SR Policy

- A maximum of 500 SR policies are supported.
- Only native COE is supported.
- Effective Cisco IOS XE Bengaluru 17.5.1, Bandwidth optimization based on SR tactical policy is supported on RSP3.
- Bandwidth optimization by using COE is not supported.
- PIC core is not supported over SR-TE tunnel.
- PIC edge over SR-TE is not supported.
- Effective Cisco IOS XE Bengaluru 17.5.1, ECMP over SR-TE is supported on RSP3.
- 6PE and 6VPE are not supported with three and four transport labels.
- IPv6 is not supported.
- A maximum of 10,000 VPNv4 prefix limits are supported.

- BGP LU (RFC 3107) is not supported for intra-AS and inter-AS.

ECMP Over SR-TE

Table 47: Feature History

Feature Name	Release Information	Feature Description
ECMP over SR-TE Policy	Cisco IOS XE Bengaluru 17.5.1	This feature allows you to configure ECMP over SR-TE policies. In case of multiple paths, this feature enables mitigation of local congestion through load balancing. This feature is supported only on Cisco ASR 900 RSP3 module.

The following sections explain how local congestion can be mitigated and how ECMP can be deployed over SR-TE policies to attain load balancing.

Note The traffic that is load balanced over multiple paths is HW-load balanced.

Restrictions for ECMP over SR-TE Policies

Cisco ASR 900 RSP3 module supports **sr_5_label_push_enable** and **sr_pfp_enable** templates. Following restrictions apply for different template combinations.

With **sr_5_label_push_enable** template:

- Only one service label is supported with LB over SR-TE tunnels with three or four TE labels. This service label includes L3VPN, L2VPN, 6PE, 6VPE, and RFC 3107 BGP-LU label.
- 6PE and 6VPE are not supported with three and four SR-TE tunnel labels.
- Segment routing is not supported in **enable_portchannel_qos_multiple_active** template.
- HW load balancing for L2VPN/EVPN services is not supported if the L2VPN/EVPN destination has a static route configured over SR-TE tunnel.

With **sr_pfp_enable** template:

- SR PM HW time stamping is not supported.
- VLAN COS marking is not supported.
- HW load balancing is not supported.
- Policer based hierarchical QOS on the ingress is not supported.
- Short-Pipe tunneling mode is not supported.

Other Restrictions:

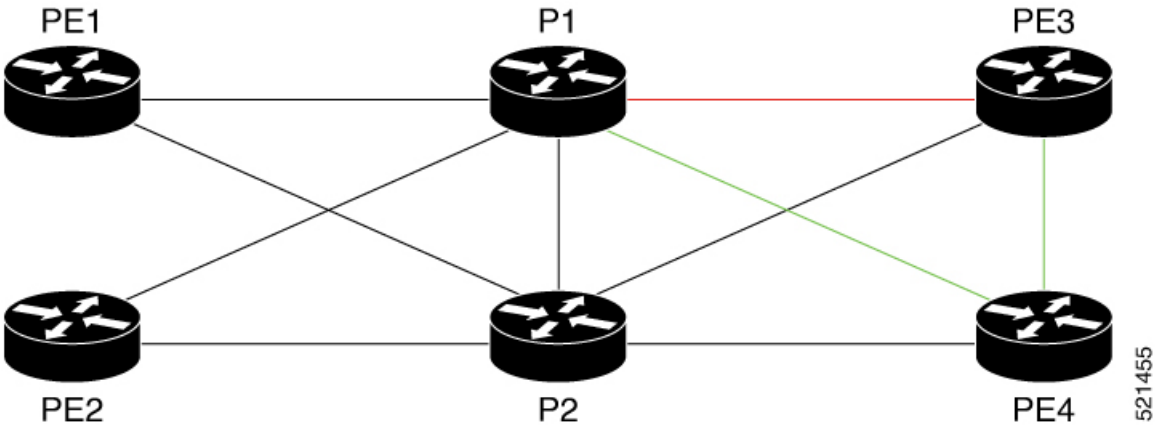
- ECMP over SR-TE is not supported with COE.
- PIC core over SR-TE tunnels are not supported.
- PIC edge over SR TE tunnels are not supported.
- PIC edge multipath over SR TE tunnels are not supported.
- W-ECMP is not supported.
- Next hop ECMP is not supported within an SR policy.
- Local congestion mitigation (LCM) is applicable only for best effort traffic. All other delay sensitive traffic uses safe SIDs (Flex Algo 128). Delay sensitive traffic is not redirected using the LCM tunnels.

Local Congestion Mitigation

In today's network deployments it is important for every router in the network to have the capability to provision the traffic in such a way that it avoids the congestion based on the amount of traffic ingressing and egressing out of it. In order to provision this congestion mitigation, it is essential for the routers to support Equal Cost Multi-Path (ECMP) load balancing, that is, distributing the traffic based on the number of paths available to reach the destination.

Congestion mitigation helps the routers to move certain traffic to a different path than the current path, using the tactical SR policies. When the link congestion threshold is crossed, the COE (Cisco Optimization Engine) that monitors the link congestion based on the interface counters, pushes these tactical policies using PCE. These PCE initiated tactical policies that are used for local congestion mitigation (LCM) are deployed when necessary and only best effort traffic is load balanced over these tactical SR-TE policies.

Figure 39: Illustration of Local Congestion Mitigation



In the above topology, let us assume that the best effort traffic is coming in to P1 from PE1 and PE2 for the destination PE3 and the link between P1 and PE3 is congested. To mitigate the congestion between P1 and PE3, ECMP paths from P1 and PE3 are required. With segment routing this is achieved by deploying multiple tactical SR policies from P1 to PE3, one through directly connected link P1-PE3 and the other through the path P1-PE4-PE3. These policies are called tactical policies and are used to avoid local congestion mitigation by load balancing the best effort traffic over these tactical policies. The LCM is applicable only for best effort traffic. All other delay sensitive traffic would use safe SIDs (Flex Algo 128). Delay sensitive traffic is not

redirected using the LCM tunnels. Originating traffic is directed on non-LCM tunnels and transiting traffic with safe-SIDs is treated as normal label entry traffic and forwarded accordingly.

In the above topology, any node may deploy LCM tactical tunnels to mitigate congestion over a particular link. These nodes transit or sometimes originate the traffic to the LCM tunnel end points or even beyond the tunnel end points.

Let us assume that PE nodes originate the traffic and P nodes are transit node for the traffic originated somewhere else. Based on these combinations following are the different types of traffic that have to be considered:

As a PE Node,

- L3VPN best effort traffic
- L2VPN best effort traffic
- Global traffic

As a P node,

- Any traffic that comes in for a non-flexible algorithm 0 label is treated as an entry swap on the Label lookup.
- Any traffic that comes in for flexible algorithm 0 label is treated as a swap case or it may be translated to pop and push stack of labels, if there is an LCM created for that outgoing link based on congestion.

Based on the number of TE labels that the LCM tunnels have to push, the number of labels outside of TE labels can be either one or two (service labels).

Load Balancing

At the head end, following are the different types of traffic that is subjected to load balancing. The traffic type here includes both best effort and delay sensitive.

As a PE Node,

- L3VPN traffic
- L2VPN traffic
- Global traffic

As a P node,

- Any traffic that comes in is treated based on the Label lookup.

Autoroute Announcement

Autoroute announcement or bandwidth optimization is used to steer traffic away from congested links and better utilize the network.

The PCEP message contains SID list to be deployed by the head-end. Path Computation Client (PCC) profiles allow autoroute announce to be activated for the policy instantiated by PCEP, using the profile IDs. For example, if the PCE instantiates a policy with profile ID 1 and the head-end where the policy is being instantiated has the PCC profile ID 1 configured with autoroute announce, PCE initiated SR policy is activated for that policy.

Autoroute announce can be configured under both policies created with strict SID and policies created with non-strict SID. The main difference between configuring autoroute under policies created with strict SID (assume A) and non-strict SID (assume B) is that with A, the lookup entry will be programmed only in RIB whereas with B, the lookup entry will be programmed in RIB and LFIB for flexible algorithm label 0.

Static Route Configuration

By adding a static route to the same destination but with different tunnels having the same endpoint, a load balancing is formed for the route over the tunnels configured. This is applicable for all types of traffic.

Next Hop ECMP within a SR Policy

If there is a SR policy created to a destination with a set of SIDs and the SR policy headend have multiple equal paths to reach the next hop, no ECMP is formed to reach the next hop within the SR policy.

Configuring with IGP Autoroute Announce

```
pce
    address ipv4 10.13.13.13
segment-routing traffic-eng
   peer ipv4 10.1.1.1
    segment-list name ss1

    policy 100
     binding-sid mpls 15999
     color 100 end-point ipv4 10.12.12.12
     candidate-paths
      preference 10
       dataplane mpls
 profile-id 100
```

Now, to push the PCE initiated OSPF autoroute announce from PCE to PCC, the profile IDs on PCE and PCC must match. The below configuration shows the PCC configuration and that the profile ID is matching with PCE and thus the autoroute announce is enabled.

```
segment-routing traffic-eng
  pcc
  pce address 10.13.13.13 source-address 10.1.1.1
  profile 100
   autoroute
    include all
```

Verifying SR Policy with Autoroute Announce

```
ASR903-R1#show segment-routing traffic-eng policy all

Name: *10.12.12.12|100 (Color: 100 End-point: 10.12.12.12)
Owners : PCEP
Status:
Admin: up, Operational: up for 66:41:16 (since 09-18 16:56:50.444)
Candidate-paths:
Preference 10 (PCEP):
PCC profile: 100
Dynamic (pce 10.13.13.13) (active)
Metric Type: TE, Path Accumulated Metric: 5
16003 [Prefix-SID, 10.3.3.3]
16012 [Prefix-SID, 10.12.12.12]
Attributes:
Binding SID: 15999
```

```
Allocation mode: explicit
State: Programmed
Autoroute:
Include all
```

Verifying ISIS Autoroute for IGP

Use the following two commands to verify the ISIS Autoroute for IGP:

```
ASR903-R1#show ip cef 10.12.12.12 -------------IGP ROUTE
10.12.12.12/32
nexthop 10.12.12.12 Tunnel65536 ------------Tunnel pushed for IGP ROUTE
```

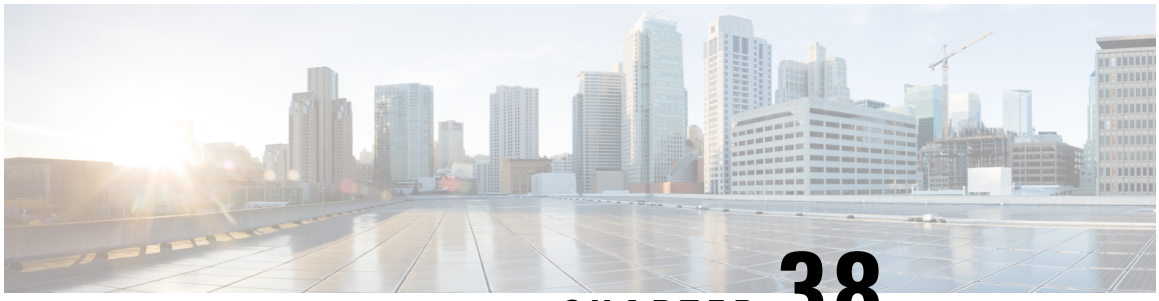
```
ASR903-R1# show ip cef 10.12.12.12 internal
10.12.12.12/32, epoch 3, RIB[I], refcnt 6, per-destination sharing
  sources: RIB
  feature space:
    IPRM: 0x00028000
    Broker: linked, distributed at 1st priority
    LFD: 10.12.12.12/32 0 local labels
        contains path extension list
  ifnums:
    Tunnel65536(64)
  path list 3C97B678, 3 locks, per-destination, flags 0x49 [shble, rif, hwcn]
    path 3E393010, share 1/1, type attached nexthop, for IPv4
     MPLS short path extensions: [rib | lblmrg | srlbl] MOI flags = 0x1 label implicit-null

      nexthop 10.12.12.12 Tunnel65536, IP midchain out of Tunnel65536 2FFE3D00
  output chain:
    IP midchain out of Tunnel65536 2FFE3D00
    label [16012|16012]
    FRR Primary (0x3D9D4CE0)
      <primary: TAG adj out of Port-channel1, addr 10.100.0.2 3C9559C0>
      <repair:  TAG adj out of BDI1110, addr 10.111.0.2 3C954FC0>
```

Verify the Tunnel ID on the SR Policy

```
ASR903-R1# show segment-routing traffic-eng policy name margin detail
Name: Margin (Color: 1000 End-point: 10.12.12.12)
  Owners : CLI
  Status:
    Admin: up, Operational: up for 00:50:52 (since 09-16 11:00:06.697)
  Candidate-paths:
    Preference 10 (CLI):
      Dynamic (pce 10.13.13.13) (active)
        Metric Type: TE, Path Accumulated Metric: 5
          16012 [Prefix-SID, 10.12.12.12]
  Attributes:
    Binding SID: 15900
      Allocation mode: explicit
      State: Programmed
  IPv6 caps enabled
  Tunnel ID: 65536 (Interface Handle: 0x15B)
  Per owner configs:
    CLI
      Binding SID: 15900
  Stats:
    Packets: 535473  Bytes: 805338440
Event history:
    Timestamp           Client          Event type          Context: Value
    ---------           ------          ----------          -------: -----
    09-16 11:00:06.377  CLI             Policy created      Name: CLI
    09-16 11:00:06.418  CLI             Set colour          Colour: 1000
```

```
09-16 11:00:06.418   CLI             Set end point        End-point: 10.12.12.12
09-16 11:00:06.446   CLI             Set binding SID      BSID: Binding SID set
09-16 11:00:06.577   CLI             Set dynamic          Path option: dynamic
09-16 11:00:06.620   CLI             BSID allocated       FWD: label 15900
09-16 11:00:06.637   FH Resolution   Policy state UP      Status: PATH RESOLVED
09-16 11:00:06.697   FH Resolution   Policy state DOWN    Status: PATH NOT RESOLVED
09-16 11:00:06.706   CLI             Set dynamic pce      Path option: dynamic pce
09-16 11:00:07.240   FH Resolution   Policy state UP      Status: PATH RESOLVED
09-16 11:00:09.520   FH Resolution   REOPT triggered      Status: REOPTIMIZED
```



CHAPTER 38

DC-PE Router in Cisco ACI to SR-MPLS Hand-off

SR-MPLS Hand-off is an interconnection option that enables Cisco ACI to WAN interconnect using Segment Routing (SR) MPLS underlay.

From Cisco IOS XE 17.14.1a, Cisco ASR 1000 Series Aggregation Services Routers and Cisco Catalyst 8500 Series Edge Platforms can be used as intermediate DC-PE devices in an ACI to SR-MPLS Hand-off interconnection.

- Prerequisites, on page 347
- Restrictions, on page 347
- Information About DC-PE Router in ACI to SR-MPLS Hand-off , on page 347
- Supported Platforms , on page 348
- How to Configure the DC-PE Router , on page 348
- Verifying DC-PE Router Configuration , on page 356
- Troubleshooting and Debugging , on page 359
- Feature Information for DC-PE Router in Cisco ACI to SR-MPLS Hand-off, on page 360

Prerequisites

There are no specific prerequisites for DC-PE Router in ACI to SR-MPLS Hand-off.

Restrictions

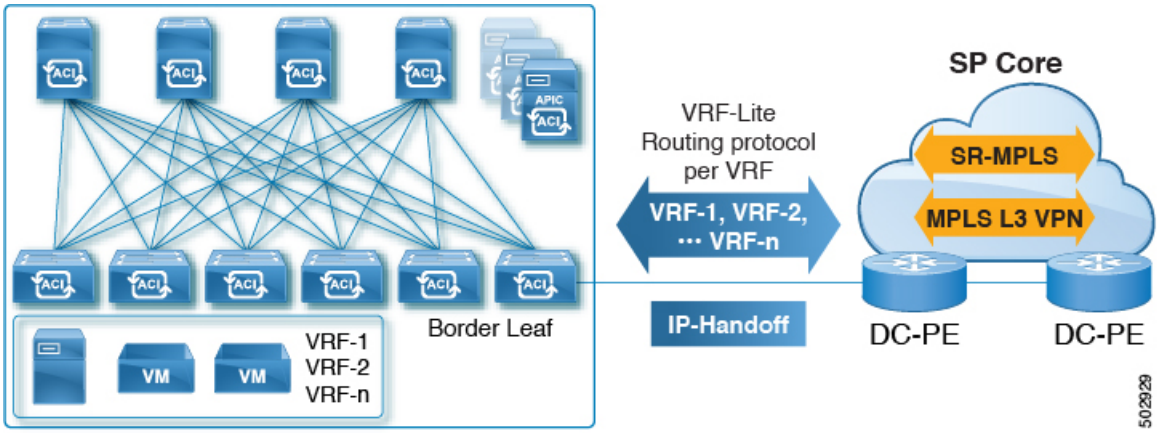
- iBGP is not supported between DC-PE and border/remote leaf.
- The router ID must be unique across all border leaf switches and the DC-PE.

Information About DC-PE Router in ACI to SR-MPLS Hand-off

SR/MPLS Handoff is an interconnection option that enables Cisco ACI fabric to WAN interconnect using Segment Routing (SR) MPLS underlay. SR/MPLS is a better solution than others known solution as it is much more common for an SP core. The solution brings the following benefits:

- Unified transport and policies between DC and SP

- Single Control Plane session for multiple VRFs
- Traffic engineering in the SP core controlled from the DC



For more information about Cisco ACI fabric and the underlying ACI to SR-MPLS hand-off interconnection, see the following publications:

- ACI SRMPLS Handoff Whitepaper
- ACI SRMPLS Architecture:
  1. Validated Design for Cisco ACI to SR-MPLS Handoff - Introduction
  2. Validated Design for Cisco ACI to SR-MPLS Handoff - Tenant Configuration
  3. ACI Fabric L3Out White Paper
- ACI SRMPLS Architecture/ Sample Use Cases

Supported Platforms

From Cisco IOS XE 17.14.1a, the following routers can be configured as DC-PE device in an ACI to SR-MPLS hand-off interconnection:

- Cisco ASR 1000 Series Aggregation Services Routers
- Cisco Catalyst 8500 Series Edge Platforms

How to Configure the DC-PE Router

Perform the following steps to configure the VRF and BGP on the DC-PE router.

Configuring VRF on the DC-PE Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *vpn-route-distinguisher*
5. **address-family ipv4** [**multicast** | **unicast**]
6. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
7. **route-target** {**export** | **import** | **both**} *route-target-ext-community* **stitching**
8. **exit-address-family**
9. **address-family ipv6** [**multicast** | **unicast**]
10. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
11. **route-target** {**export** | **import** | **both**} *route-target-ext-community* **stitching**
12. **exit-address-family**
13. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. Enter password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition test`	Enters the VRF configuration mode for the specified VRF instance.
Step 4	**rd** *vpn-route-distinguisher* **Example:** `Device(config-vrf)#  rd 65000:1`	Specifies the route distinguisher for the VRF instance.
Step 5	**address-family ipv4** [**multicast**	**unicast**] **Example:** `Device(config-vrf)# address-family ipv4`	Enters the IPv4 address family configuration mode.
Step 6	**route-target** {**export**	**import**	**both**} *route-target-ext-community*	Creates a list of import, export, or both import and export route target communities for the specified VRF.

	Command or Action	Purpose

	Example:
	`Device(config-vrf-af)# route-target import 1:1`
	Example:
	`Device(config-vrf-af)# route-target export 2:2`
		Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y).

Step 7	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **stitching**
	Example:
	`Device(config-vrf-af)# route-target import 3:3 stitching`
	Example:
	`Device(config-vrf-af)# route-target export 4:4 stitching`
		Configures importing, exporting, or both importing and exporting of EVPN route target communities for the VRF.

Step 8	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 9	**address-family ipv6** [**multicast** | **unicast**] **Example:** `Device(config-vrf)# address-family ipv6`	Enters the IPv6 address family configuration mode.

Step 10	**route-target** {**export** | **import** | **both**} *route-target-ext-community*
	Example:
	`Device(config-vrf-af)# route-target import 1:1`
	Example:
	`Device(config-vrf-af)# route-target export 2:2`
		Creates a list of import, export, or both import and export route target communities for the specified VRF.
		Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y).

Step 11	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **stitching**
	Example:
	`Device(config-vrf-af)# route-target import 3:3 stitching`
	Example:
	`Device(config-vrf-af)# route-target export 4:4 stitching`
		Configures importing, exporting, or both importing and export of EVPN route target communities for the VRF.

Step 12	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 13	**end** **Example:** `Device(config-vrf)# end`	Returns to privileged EXEC mode.

Example

The following example demonstrates the VRF configuration required for the DC-PE router:

```
vrf definition test
  rd 65000:1
  address-family ipv4
            route-target import 1:1
            route-target export 2:2
            route-target import 3:3 stitching
            route-target export 4:4 stitching
  exit
  address-family ipv6
            route-target import 1:1
            route-target export 2:2
            route-target import 3:3 stitching
            route-target export 4:4 stitching
  exit
```

Configuring BGP on the DC-PE router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **routerbgp** *as-number*
4. **neighbor** *dc-border-leaf-address* **remote-as** *number*
5. **neighbor** *wan-router-address* **remote-as** *number*
6. **address-family l2vpn evpn**
7. **import vpnv4 unicast** [**re-originate**]
8. **import vpnv6 unicast** [**re-originate**]
9. **neighbor** *ip-address* **activate**
10. **neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**]
11. **exit-address-family**
12. **address-family vpnv4**
13. **import l2vpn evpn** [**re-originate**]
14. **neighbor** *ip-address* **activate**
15. **neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**]
16. **neighbor** {**ip-address** | **peer-group-name**} **next-hop-self** [**all**]
17. **exit-address-family**
18. **address-family vpnv6**
19. **import l2vpn evpn** [**re-originate**]
20. **neighbor** *ip-address* **activate**
21. **neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**]
22. **neighbor** {**ip-address** | **peer-group-name**} **next-hop-self** [**all**]
23. **exit-address-family**
24. **address-family ipv4 vrf** *vrf-name*
25. **maximum-paths eibgp** *number*

26. **exit-address-family**
27. **address-family ipv6 vrf** *vrf-name*
28. **maximum-paths eibgp** *number*
29. **exit-address-family**
30. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**routerbgp** *as-number* **Example:** `Device(config)# router bgp 1`	Configures a BGP routing process and enters router configuration mode.
Step 4	**neighbor** *dc-border-leaf-address* **remote-as** *number* **Example:** `Device(config-router)# neighbor 1.1.1.1 remote-as 2`	Defines multiprotocol-BGP neighbors in the EVPN network. Use the IP address of the spine switch as the neighbor IP address. This configures the spine switch as a BGP neighbor.
Step 5	**neighbor** *wan-router-address* **remote-as** *number* **Example:** `Device(config-router)# neighbor 2.2.2.2 remote-as 1`	Defines multiprotocol-BGP neighbors in the external MPLS network. Use the IP address of the external MPLS network peer as the neighbor IP address. This configures the external MPLS network peer as a BGP neighbor.
Step 6	**address-family l2vpn evpn** **Example:** `Device(config-router)# address-family l2vpn evpn`	Specifies the L2VPN address family and enters address family configuration mode.
Step 7	**import vpnv4 unicast** [**re-originate**] **Example:** `Device(config-router-af)# import vpnv4 unicast re-originate`	Reoriginates the VPNv4 routes imported from the external peer into the EVPN address family as EVPN routes, and distributes within the EVPN fabric.

	Command or Action	Purpose
Step 8	**import vpnv6 unicast** [**re-originate**] **Example:** `Device(config-router-af)# import vpnv6 unicast re-originate`	Reoriginates the VPNv6 routes imported from the external peer into the EVPN address family as EVPN routes, and distributes within the EVPN fabric.
Step 9	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 1.1.1.1 activate`	Enables the exchange information from a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 10	**neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 1.1.1.1 send-community both`	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address. **Note** Use either extended or both keywords. E keyword.
Step 11	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 12	**address-family vpnv4** **Example:** `Device(config-router)# address-family vpnv4`	Specifies the VPNv4 address family and enters address family configuration mode.
Step 13	**import l2vpn evpn** [**re-originate**] **Example:** `Device(config-router-af)# import l2vpn evpn re-originate stitching-rt`	Reoriginates the EVPN routes imported from the EVPN fabric into the VPNv4 address family as VPNv4 routes and distributes them to the external network.
Step 14	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 2.2.2.2 active`	Enables the exchange information from a BGP neighbor. Use the IP address of the external MPLS network router as the neighbor IP address.
Step 15	**neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 2.2.2.2 send-community both`	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the external MPLS network router as the neighbor IP address. **Note** Use either extended or both keywords. E keyword.

	Command or Action	Purpose
Step 16	**neighbor** {**ip-address** | **peer-group-name**} **next-hop-self** [**all**] **Example:** `Device(config-router-af)# neighbor 2.2.2.2 next-hop-self all`	Configures the router as the next hop for a BGP-speaking neighbor or peer group. The all keyword is mandatory when implementing external connectivity through iBGP, where the EVPN fabric and the MPLS network are in the same BGP autonomous system number. The all keyword is optional when implementing external connectivity through eBGP, where the EVPN fabric and the MPLS network are in different BGP autonomous system numbers
Step 17	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 18	**address-family vpnv6** **Example:** `Device(config-router)# address-family vpnv6`	Specifies the VPNv6 address family and enters address family configuration mode.
Step 19	**import l2vpn evpn** [**re-originate**] **Example:** `Device(config-router-af)# import l2vpn evpn re-originate stitching-rt`	Reoriginates the EVPN routes imported from the EVPN fabric into the VPNv6 address family as VPNv6 routes and distributes them to the external network.
Step 20	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 2.2.2.2 active`	Enables the exchange information from a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 21	**neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 2.2.2.2 send-community both`	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address. **Note** Use either extended or both keywords. Exte keyword.
Step 22	**neighbor** {**ip-address** | **peer-group-name**} **next-hop-self** [**all**] **Example:** `Device(config-router-af)# neighbor 2.2.2.2 next-hop-self all`	Configures the router as the next hop for a BGP-speaking neighbor or peer group. The all keyword is mandatory when implementing external connectivity through iBGP, where the EVPN fabric and the MPLS network are in the same BGP autonomous system number.

	Command or Action	Purpose
		The all keyword is optional when implementing external connectivity through eBGP, where the EVPN fabric and the MPLS network are in different BGP autonomous system numbers
Step 23	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 24	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf test`	Places the router in address family configuration mode. Separate VRF multipath configurations are isolated by unique route distinguisher.
Step 25	**maximum-paths eibgp** *number* **Example:** `Device(config-router-af)# maximum-paths eibgp 16`	Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table. **Note** You can configure the **maximum-paths** mode
Step 26	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 27	**address-family ipv6 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv6 vrf test`	Places the router in address family configuration mode. Separate VRF multipath configurations are isolated by unique route distinguisher.
Step 28	**maximum-paths eibgp** *number* **Example:** `Device(config-router-af)# maximum-paths eibgp 16`	Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table. **Note** You can configure the **maximum-paths** mode
Step 29	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 30	**end** **Example:** `Device(config-vrf)# end`	Returns to privileged EXEC mode.

Example

The following example demonstrates the VRF configuration required for the DC-PE router:

```
router bgp 1
      neighbor 1.1.1.1 remote-as 2
      neighbor 2.2.2.2 remote-as 1
      address-family l2vpn evpn
                import vpnv4 unicast re-originate
                import vpnv6 unicast re-originate
                neighbor 1.1.1.1 active
                neighbor 1.1.1.1 send-community both
      exit
      address-family vpnv4
                import l2vpn evpn re-originate stitching-rt
                neighbor 2.2.2.2 active
                neighbor 2.2.2.2 send-community both
                neighbor 2.2.2.2 next-hop-self all
      exit
      address-family vpnv6
                import l2vpn evpn re-originate stitching-rt
                neighbor 2.2.2.2 active
                neighbor 2.2.2.2 send-community both
                neighbor 2.2.2.2 next-hop-self all
      exit
      address-family ipv4 vrf test
        maximum-paths eibgp 16
      exit
      address-family ipv6 vrf test
        maximum-paths eibgp 16
      exit
```

Verifying DC-PE Router Configuration

This section provides the show commands that can be used to verify the DC-PE router configuration.

Verifying IPv4 and IPv6 Route from ACI

Use the following commands to verify IPv4 route from ACI:

```
Router#show bgp l2vpn evpn route-type 5 0 99.1.2.0 24
BGP routing table entry for [5][2:2][0][24][99.1.2.0]/17, version 2
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 1
  65000 65001
   2.2.2.2 (via default) from 5.5.5.5 (5.5.5.5)
    Origin incomplete, localpref 100, valid, external, best
    EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, VNI  Label 0, MPLS VPN Label
 19
    Extended Community: RT:2:2 Color:10
    rx pathid: 0, tx pathid: 0x0
    Updated on Feb 27 2024 15:46:31 PST


Router#show bgp vpnv4 uni all 99.1.2.0
```

```
BGP routing table entry for 6:6:99.1.2.0/24, version 2
Paths: (1 available, best #1, table red)
  Advertised to update-groups:
   1
  Refresh Epoch 1
  65000 65001, imported path from [5][2:2][0][24][99.1.2.0]/17   (global)
    2.2.2.2 (via default) from 5.5.5.5 (5.5.5.5)
     Origin incomplete, localpref 100, valid, external, best
     Extended Community: RT:2:2 Color:10
     mpls labels in/out IPv4 VRF Aggr:19/19
     rx pathid: 0, tx pathid: 0x0
     Updated on Feb 27 2024 15:46:31 PST


Router#show ip route vrf red 99.1.2.0
Routing Table: red
Routing entry for 99.1.2.0/24
Known via "bgp 65100", distance 20, metric 0
Tag 65000, type external
Last update from 2.2.2.2 00:07:23 ago
Routing Descriptor Blocks:
* 2.2.2.2 (default), from 5.5.5.5, 00:07:23 ago
opaque_ptr 0x7F055237F160
Route metric is 0, traffic share count is 1
AS Hops 2
Route tag 65000
MPLS label: 19
```

Use the following commands to verify IPv6 route from ACI:

```
Router#show bgp l2vpn evpn route-type 5 0 2001::99:1:2:0 112
BGP routing table entry for [5][2:2][0][112][2001::99:1:2:0]/29, version 4
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 1
  65000 65001
   2.2.2.2 (via default) from 5.5.5.5 (5.5.5.5)
    Origin incomplete, localpref 100, valid, external, best
    EVPN ESI: 00000000000000000000, Gateway Address:::,VNI Label 0,MPLS VPN Label 21
    Extended Community: RT:2:2 Color:10
    rx pathid: 0, tx pathid: 0x0
    Updated on Feb 27 2024 15:46:31 PST


Router#show bgp vpnv6 uni all 2001::99:1:2:0/112
BGP routing table entry for [6:6]2001::99:1:2:0/112, version 2
Paths: (1 available, best #1, table red)
  Advertised to update-groups:
   1
  Refresh Epoch 1
  65000 65001, imported path from [5][2:2][0][112][2001::99:1:2:0]/29 (global)
   ::FFFF:2.2.2.2 (via default) from 5.5.5.5 (5.5.5.5)
     Origin incomplete, localpref 100, valid, external, best
     Extended Community: RT:2:2 Color:10
     mpls labels in/out IPv6 VRF Aggr:20/21
     rx pathid: 0, tx pathid: 0x0
     Updated on Feb 27 2024 15:46:31 PST


Router#show ipv6 route vrf red 2001::99:1:2:0/112
Routing entry for 2001::99:1:2:0/112
Known via "bgp 65100", distance 20, metric 0
Tag 65000, type external
Route count is 1/1, share count 0
Routing paths:
```

```
2.2.2.2%default indirectly connected
Route metric is 0, traffic share count is 1
MPLS label: 21
From ::FFFF:5.5.5.5
opaque_ptr 0x7F05523C42C8
Last updated 00:10:33 ago
```

Verifying IPv4 and IPv6 Route from WAN

Use the following commands to verify IPv4 route from WAN:

```
Router#show bgp vpnv4 uni vrf red 13.13.13.13
BGP routing table entry for 6:6:13.13.13.13/32, version 19
Paths: (1 available, best #1, table red)
Flag: 0x100
  Not advertised to any peer
  Refresh Epoch 1
  65013, imported path from 12:12:13.13.13.13/32 (global)
    12.12.12.12 (metric 30) (via default) from 7.7.7.7 (7.7.7.7)
     Origin incomplete, metric 0, localpref 100, valid, internal,  best
     Extended Community: RT:12:12 Color:10
     Originator: 12.12.12.12, Cluster list: 7.7.7.7
     mpls labels in/out nolabel/18
     binding SID: 22 (color - 10) (state - UP)
     rx pathid: 0, tx pathid: 0x0
     Updated on Feb 27 2024 15:46:32 PST
```

```
Router#show bgp l2vpn evpn route-type 5 0 13.13.13.13 32
BGP routing table entry for [5][6:6][0][32][13.13.13.13]/17, version 18
Paths: (1 available, best #1, table EVPN-BGP-Table)
Advertised to update-groups:
   1
  Refresh Epoch 1
  65013, imported path from base
   12.12.12.12 (metric 30) (via default) from 7.7.7.7 (7.7.7.7)
   Origin incomplete, metric 0, localpref 100, valid, internal, best
   EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, local   vtep: 0.0.0.0, VNI
Label 0, MPLS VPN Label 18, MPLS VPN Local Label  19
   Extended Community: RT:2:2 RT:4:4 Color:10
   Originator: 12.12.12.12, Cluster list: 7.7.7.7
   rx pathid: 0, tx pathid: 0x0
   Updated on Feb 27 2024 15:46:32 PST
```

```
Router#show ip route vrf red 13.13.13.13
Routing Table: red
Routing entry for 13.13.13.13/32
Known via "bgp 65100", distance 200, metric 0
Tag 65013, type internal
Routing Descriptor Blocks:
* Binding Label: 22, from 7.7.7.7, 00:07:48 ago
opaque_ptr 0x7F055237ED70
Route metric is 0, traffic share count is 1
AS Hops 1
Route tag 65013
MPLS label: 18
MPLS Flags: MPLS Required
```

Use the following commands to verify IPv6 route from WAN:

```
Router#show bgp vpnv6 uni vrf red 2001::13:13:13:13/128
```

```
BGP routing table entry for [6:6]2001::13:13:13:13/128, version 19
Paths: (1 available, best #1, table red)
Flag: 0x100
  Not advertised to any peer
  Refresh Epoch 1
  65013, imported path from [12:12]2001::13:13:13:13/128 (global)
   ::FFFF:12.12.12.12 (metric 30) (via default) from 7.7.7.7 (7.7.7.7)
   Origin incomplete, metric 0, localpref 100, valid, internal, best
   Extended Community: RT:12:12 Color:10
   Originator: 12.12.12.12, Cluster list: 7.7.7.7
   mpls labels in/out nolabel/20
   binding SID: 22 (color - 10) (state - UP)
   rx pathid: 0, tx pathid: 0x0
   Updated on Feb 27 2024 15:46:32 PST


Router#show bgp l2vpn evpn route-type 5 0 2001::13:13:13:13 128
BGP routing table entry for [5][6:6][0][128][2001::13:13:13:13]/29, version 12
Paths: (1 available, best #1, table EVPN-BGP-Table)
   Advertised to update-groups:
     1
   Refresh Epoch 1
   65013, imported path from base
   ::FFFF:12.12.12.12 (metric 30) (via default) from 7.7.7.7 (7.7.7.7)
     Origin incomplete, metric 0, localpref 100, valid, internal, best
     EVPN ESI: 00000000000000000000, Gateway Address: ::, local vtep: 0.0.0.0, VNI Label
0, MPLS VPN Label 20, MPLS VPN Local Label 20
     Extended Community: RT:2:2 RT:4:4 Color:10
     Originator: 12.12.12.12, Cluster list: 7.7.7.7
     rx pathid: 0, tx pathid: 0x0
     Updated on Feb 27 2024 15:46:32 PST


Router#show ipv6 route vrf red 2001::13:13:13:13/128
Routing entry for 2001::13:13:13:13/128
Known via "bgp 65100", distance 200, metric 0
Tag 65013, type internal
Route count is 1/1, share count 0
Routing paths:
Bind Label: 22 indirectly connected
Route metric is 0, traffic share count is 1
MPLS label: 20
From ::FFFF:7.7.7.7
opaque_ptr 0x7F05523C3ED8
Last updated 00:10:03 ago
```

Troubleshooting and Debugging

The following debug commands can be used to enable the debugs required for debugging BGP Label Manager:

```
debug bgp lmm address-family vpnv4
debug bgp lmm address-family vpnv6
```

The following example shows the output of the **debug bgp lmm address-family vpnv4/6** command:

```
*Jul 18 21:32:09.835: BGP_LMM (VPNv4): Add update info for 1:1:3.3.3.0/24, neighbor 1.1.1.3,
 NH  unchanged (no), topology neighbor labeled (yes)
*Jul 18 21:34:48.577: BGP_LMM (VPNv6): Add update info for [1:1]3333::/120, neighbor 1.1.1.3,
 NH unchanged (no), topology neighbor labeled (yes)
*Jul 18 21:32:09.835: BGP_LMM (VPNv4): Allocated and installed a per VRF aggregate label
```

```
10 for vrf  red, address family ipv4"
*Jul 18 21:32:09.835: BGP_LMM (VPNv4): Allocated and installed a per VRF aggregate label
11 for vrf  red, address family ipv6"
```

The following debug commands can be used to debug BGP EVPN to L3VPN import/re-origination:

```
debug bgp all import updates
debug bgp all import events
```

The following example shows the output of the **debug bp all import** command:

```
*Jul 21 14:31:22.693: BGP VPN-IMP: red:VPNv4 Unicast:base 1:1:3.3.3.0/24 Exporting doing
PATHS.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base Building ETL from VPN
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base GBL Building ETL.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base -> global:IPv4 Unicast:base Creating
 Import Topo.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base -> global:IPv4 Unicast:base GBL Adding
 topology IPv4 Unicast to ETL.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base -> global:IPv4 Multicast:base Creating
 Import Topo.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base -> global:IPv4 Multicast:base GBL
Adding to ETL.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base Building GBL ETL done.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base L2VPN E-VPN AF_PRIV Building ETL.
```

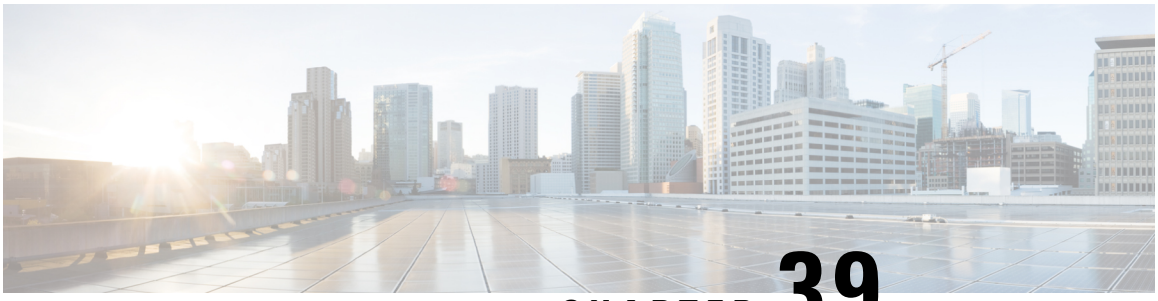
Feature Information for DC-PE Router in Cisco ACI to SR-MPLS Hand-off

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 48: Feature Information for DC-PE Router in Cisco ACI to SR-MPLS Hand-off

Feature Name	Releases	Feature Information
DC-PE Router in Cisco ACI to SR-MPLS Hand-off	Cisco IOS XE 17.14.1a	From Cisco IOS XE 17.14.1a, Cisco ASR 1000 Series Aggregation Services Routers and Cisco Catalyst 8500 Series Edge Platforms can be used as intermediate DC-PE devices in Cisco ACI to SR-MPLS hand-off interconnection. SR-MPLS hand-off is an interconnection option that enables Cisco ACI to WAN interconnect using Segment Routing (SR) MPLS underlay.



CHAPTER 39

Segment Routing over IPv6

Segment Routing (SR) can be applied on both MPLS and IPv6 dataplanes. From Cisco IOS XE 17.12.1a, Segment Routing over IPv6 (SRv6) extends Segment Routing support over the IPv6 dataplane.

- Segment Routing over IPv6, on page 362
- Configuring SRv6, on page 366
- SRv6 under IS-IS, on page 370
- SRv6 BGP-Based Services, on page 372
- EVPN VPWS over SRv6, on page 385
- BGP SRv6 L3VPN On-Demand Next-Hop, on page 396
- SRv6 Traffic Engineering Policies, on page 406
- Performance Measurement for SRv6, on page 413
- SRv6 OAM, on page 419

Segment Routing over IPv6

Feature Information

Table 49: Feature Information Table for SRv6

Feature Name	Release	Description
Segment Routing over IPv6 Dataplane	Cisco IOS XE Release 17.12.1a	Segment Routing (SR) can currently be applied on Multiprotocol Label Switching (MPLS) dataplane. From Cisco IOS XE 17.12.1a, SR is supported over the IPv6 dataplane for the following protocols: • Interior Gateway Protocol (IS-IS only) • Border Gateway Protocol (BGP) In addition, the following functionalities are available for Segment Routing over IPv6 dataplane: • Segment Routing Traffic Engineering Policies • Static Routes • Performance Management • Operations, Administration and Maintenance (OAM)
BGP SRv6L3VPN On-Demand Next-Hop	CiscoIOS XE Release 17.13.1a	This feature was introduced.
Enhancements to SRv6	CiscoIOS XE Release 17.16.1a	The following features were introduced: • eBGP Inter-AS • PCE-Delegated Path Computation • Enhancements to OAM Traffic Engineering

Note For information about supported platforms for each feature and release, see Supported Platforms, on page 366.

Restrictions for SRv6

- Cisco IOS XE supports uSIDs with 32-bit uSID block and 16-bit uSID IDs (3216). This format must be used for uSID locators in a SRv6 uSID domain.
- Cisco IOS XE supports up to 10 uSID locators.
- Cisco IOS XE supports the following SRv6 uSID behaviors and variants:
  - uN with PSP/USD
  - uA with PSP/USD
  - uDT4
  - uDT6
  - uDT46
- Cisco IOS XE supports **H.Encaps.Red** SRv6 policy headend behavior.

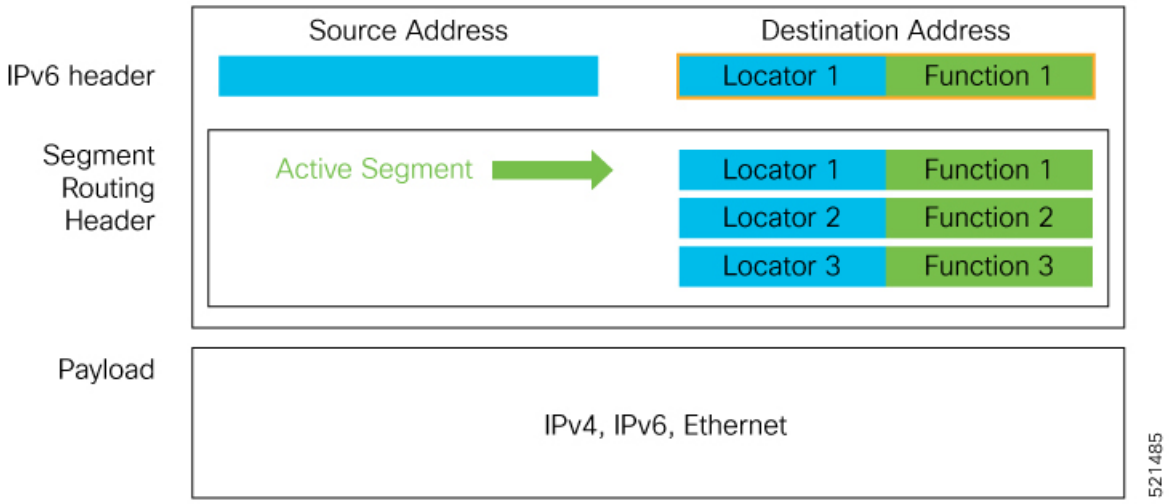
Information About SRv6

In an SR-MPLS enabled network, an MPLS label represents an instruction. The source nodes program the path to a destination in the packet header as a stack of labels.

SRv6 introduces the Network Programming framework that enables a network operator or an application to specify a packet processing program by encoding a sequence of instructions in the IPv6 packet header. Each instruction is implemented on one or several nodes in the network and identified by an SRv6 Segment Identifier (SID) in the packet. The SRv6 Network Programming framework is defined in IETF RFC 8986 SRv6 Network Programming.

In SRv6, an IPv6 address represents an instruction. SRv6 uses a new type of IPv6 Routing Extension Header, called the Segment Routing Header (SRH), in order to encode an ordered list of instructions. The active segment is indicated by the destination address of the packet, and the next segment is indicated by a pointer in the SRH.

Figure 40: Network Program in the Packet Header



The SRv6 SRH is documented in IETF RFC 8754 IPv6 Segment Routing Header (SRH).

SRv6 Node Roles

The SRv6 Node Roles are documented in the IETF RFC 8754 IPv6 Segment Routing Header (SRH).

SRv6 Head-End Behaviors

The SRv6 Head-end with Encapsulation behaviors are documented in the IETF RFC 8986 SRv6 Network Programming.

SRv6 Endpoint Behaviors

The SRv6 endpoint behaviors are documented in the IETF RFC 8986 SRv6 Network Programming.

SRv6 Endpoint Behavior Variants

The SRv6 endpoint behavior variants are documented in the IETF RFC 8986 SRv6 Network Programming.

SRv6 Micro-Segment (uSID)

Several SRv6 uSIDs may be encoded within a single 128-bit SID, called a uSID carrier.

SRv6 uSID is documented in the IETF drafts NetworkProgramming extension: SRv6 uSID instruction and Compressed SRv6 Segment List Encoding in SRH.

Throughout this chapter, SRv6 micro-segment is referred to as **uSID**.

SRv6 uSID Terminology

The SRv6 uSID terminology is documented in the Network Programming extension: SRv6 uSID instruction.

SRv6 uSID Allocation Within a uSID Block

SRv6 uSID allocations are documented in the Network Programming extension: SRv6 uSID instruction.

SRv6 Endpoint Behaviors Associated with uSID

SRv6 uSID endpoint behaviors are documented in the Network Programming extension: SRv6 uSID instruction.

SRv6 Implementation

A new command **segment-routing srv6** is introduced in Cisco IOS XE 17.12.1a to enable SRv6 configuration.

```
segment-routing srv6
    encapsulation
        source-address {ipv6-addr}
        hop-limit [propagate | <value>]
        traffic-class [propagate | <value>]
    locators
        locator <name>
          format usid-f3216
          prefix <locator-ipv6-prefix/prefix-len>
    sid holdtime <value>
    explicit-sids
        sid <SRv6-SID> behavior {end-dt46 | end-dt4 | end-dt6}
            forwarding
                path <1-1>
                    decap-and-lookup [vrf-name <vrf>]
```

The parameters for this command are described below.

SRv6 Locator Name, Prefix, and uSID-Related Parameters

This section describes the configurable keywords for the **segment-routing SRv6** command.

locator *name*	Configures the SRv6 locator.
locator *name* **prefix** *locator*	Configures the locator prefix value.
locator *name* **format** *usid-f3216*	Specifies the locator as a micro-segment (uSID).

SRv6 Encapsulation Parameters

This section describes the configurable SRv6 encapsulation parameters. These optional parameters include:

encapsulation source-address ipv6-addr	Source Address of outer encapsulating IPv6 header. The default source address for encapsulation is the lowest global unicast IPv6 address of lowest loopback interface. If loopback addresses and static encapsulation source address are not configured, the source address remains unassigned (0::0).
encapsulation hop-limit {*count* | <*propagate*>}	The hop limit of outer-encapsulating IPv6 header. The range for *count* is from 1 to 255; the default value for hop-limit is 64. Use **propagate** to set the hop-limit value by propagation (from incoming packet/frame).

encapsulation traffic-class {*value* | <*propagate*>}	The traffic-class field settings on the IPv6 header. Specify the *value* (as 2 hexadecimal nibbles) for traffic class; valid values are from 0x0 to 0xff. The default value is 0. Use **propagate** to set the traffic-class value by propagation (from incoming packet/frame).

SRv6 SID Parameters

This section describes the configurable SRv6 SID parameters.

sid holdtime *minutes*	The hold time for a stale or freed SID. The range of *minutes* is from 0 (disabled) to 60 minutes.
sid <*SRv6-SID*> **behavior** {*end-dt46* | *end-dt4* | *end-dt6*}	Configures a static SID, given the SID address and the behavior context.

Supported Platforms

From Cisco IOS XE 17.12.1a release, several SRv6 features are supported on the following platforms:

- Cisco ASR1000 RP3 + ESP100-X/ESP200-X, ASR1001-HX, ASR1002-HX
- Cisco Catalyst 8000V Edge Software
- Cisco Catalyst 8200 Series Edge platforms
- Cisco Catalyst 8300 Series Edge platforms
- Cisco Catalyst 8500 and 8500L Series Edge platforms

For information about the supported features, see Feature Information, on page 362.

From Cisco IOS XE 17.13.1a release, BGP SRv6 L3VPN ODN is supported on the following platforms:

- Cisco ASR1000 RP3 + ESP100-X/ESP200-X, ASR1001-HX, ASR1002-HX
- Cisco Catalyst 8000V Edge Software
- Cisco Catalyst 8200 Series Edge platforms
- Cisco Catalyst 8300 Series Edge platforms
- Cisco Catalyst 8500 and 8500L Series Edge platforms

Configuring SRv6

Configuring SRv6

Enabling SRv6 involves the following high-level configuration steps:

- Configuring Global SRv6 with locator(s)
- Configuring Optional SRv6 Parameters

Configuring Global SRv6 with a Locator

The following example shows how to globally enable SRv6 and configure a locator:

```
Router(config)# segment-routing srv6
Router(config-srv6)# locators
Router(config-srv6-locator)# locator myLoc1
Router(config-srv6-locator)# format usid-f3216
Router(config-srv6-locator)# prefix 2001:0:8::/48
```

Configuring Optional SRv6 Parameters

The following example shows how to configure optional SRv6 parameters:

```
Router(config)# segment-routing srv6
Router(config-srv6)# encapsulation
Router(config-srv6-encap)# source-address 1::1
Router(config-srv6-encap)# hop-limit 60
Router(config-srv6-encap)# traffic-class propagate
Router(config-srv6-encap)# exit
Router(config-srv6)# sid holdtime 10
```

Configuring SRv6 Static Explicit END.X SID

From Cisco IOS XE 17.15.1a, you can configure static explicit END.X:

```
segment-routing srv6
      explicit-sids
         sid <sid> behavior end-x-usid-psp-usd
            identification nexthop [interface] <IPv6 address> <protected>
```

Note IPv6 address is not required for point-to-point interface. Use link-local address to specify IPv6 address.

Configuring SRv6 Path MTU

From Cisco IOS XE 17.15.1a, a new global static configuration of MTU for all SRv6 policy and VPN-SID is available:

```
segment-routing srv6
  encapsulation
     mtu <1280-9960>
```

Verifying SRv6 Configuration

Use the following examples to verify SRv6 configuration.

Example 1: This example shows how to verify the locator configuration and its operational status:

```
router# show segment-routing srv6 locator
Name            Algo     Prefix                    Format              Status
----            ----     ------                    ------              ------
loc1            0        FC01:101:2::/48           usid-f3216          Up
```

Example 2: The following examples show how to view the platform capabilities and parameters:

```
router# show segment-routing srv6 capabilities-parameters
Platform Capabilities:
 SRv6:Yes
 PFP:Yes
 TILFA:No
 Endpoint behaviors:
  uN (PSP/USD)
  uA (PSP/USD)
  uDT6
  uDT4
  uDT46
  Transit.ENCAP.RED
 Encap Parameters:
  Max-SL :16
  Encap :Collapsed
  Hop-limit propagate :Yes
  Traffic-class propagate :Yes
Parameters in-use:
 Encap Parameters:
  Source Address: 2001::1:1:1:2, Loopback1 (Default)
  Hop-Limit: 64 (Default)
  Traffic-class: 0 (Default)
```

```
router# show srv6 capabilities-parameters
Platform Capabilities:
 SRv6:Yes
 PFP:Yes
 TILFA:No
 Endpoint behaviors:
  uN (PSP/USD)
  uA (PSP/USD)
  uDT6
  uDT4
  uDT46
  Transit.ENCAP.RED
 Encap Parameters:
  Max-SL :16
  Encap :Collapsed
  Hop-limit propagate :Yes
  Traffic-class propagate :Yes
Parameters in-use:
  Encap Parameters:
    Source Address: A001::1, Loopback0 (Default)
    Hop-Limit: 64 (Default)
    Traffic-class: 0 (Default)
```

Example 3: The following examples show how to view the SID overview and details:

```
router# show segment-routing srv6 sid
SID                  Locator      Behavior          Context               Owner
---                  -------      --------          -------               -----
FC01:101:2::          loc1        uN (PSP/USD)                                  SID-MGR

FC01:101:2:E000::     loc1         uDT4             ce1                           router
 bgp
FC01:101:2:E001::     loc1         uDT6             ce1                           router
 bgp
FC01:101:2:E002::     loc1        uA (PSP/USD)      Ethernet2/0 2001::99:2:3:3    router
 isis sr
FC01:101:2:E003::     loc1        uA (PSP/USD)      Ethernet2/1 2001::100:2:3:3   router
 isis sr
FC01:101:2:E004::     loc1        uA (PSP/USD)      Ethernet3/0 2001::99:2:4:4    router
 isis sr
FC01:101:2:E005::     loc1        uA (PSP/USD)      Ethernet3/1 2001::100:2:4:4   router
```

```
 isis sr
FC01:101:2:E006::     loc1         uA (PSP/USD)      Ethernet4/0 2001::99:2:5:5       router
 isis sr
FC01:101:2:E007::     loc1         uA (PSP/USD)      Ethernet4/1 2001::100:2:5:5      router
 isis sr
```

```
router# show segment-routing srv6 sid FC01:101:2:: detail
SID: FC01:101:2::   Type: DYNAMIC
 Behavior: uN (PSP/USD) (48)
 Context:
    interface: (not-set)
    vrf: (not-set), v4-topo-id: 0xFFFF, v6-topo-id: 0xFFFF
    next-hop: (not-set)
    policy: (not-set)
    distinguisher: (not-set)
 Stats:
    Packets: 0  Bytes: 0
 User list:
    User:Refcount        Locator:Refcount
    -------------        ----------------
    SID-MGR(2):1         loc1:1
 Event history:
    Timestamp            Client           Event type
    ---------            ------           ----------
    04-15 05:44:43.992   SID-MGR(2)       ALLOC
```

Example 4: The following examples show how to view stale SIDs:

```
router# show segment-routing srv6 sid stale
SID                        Locator         Behavior             Context
       Owner
---                        -------         --------             -------

FC01:101:2:E002::           loc1           uA (PSP/USD)         Ethernet2/0 2001::99:2:3:3
```

```
router# show segment-routing srv6 sid stale detail
SID: FC01:101:2:E002::
 Behavior: uA (PSP/USD) (57)
 Context:
    interface: Ethernet2/0
    vrf: (not-set), v4-topo-id: 0xFFFF, v6-topo-id: 0xFFFF
    next-hop: 2001::99:2:3:3
    policy: (not-set)
    distinguisher: (not-set)
 Event history:
    Timestamp            Client           Event type
    ---------            ------           ----------
    04-15 06:58:13.961   router isis sr(3 ALLOC
    04-15 07:24:49.831   router isis sr(3 DEALLOC
```

Example 5: The following examples show how to view the configured IPv6 route and router prefix:

```
router# show ipv6 route
(snip)
C   FC01:101:2::/48 [0/0]
     via SR0, directly connected
L   FC01:101:2::/128 [0/0]
     via SR0, receive
I2  FC01:101:3::/48 [115/10]
     via FE80::A8BB:CCFF:FE02:8F02, Ethernet2/0
     via FE80::A8BB:CCFF:FE02:8F12, Ethernet2/1
I2  FC01:101:4::/48 [115/10]
     via FE80::A8BB:CCFF:FE01:C901, Ethernet3/0
```

```
     via FE80::A8BB:CCFF:FE01:C911, Ethernet3/1
I2  FC01:101:5::/48 [115/10]
     via FE80::A8BB:CCFF:FE03:A404, Ethernet4/0
     via FE80::A8BB:CCFF:FE03:A414, Ethernet4/1
```

```
router# show ipv6 route FC01:101:2::/48
Routing entry for FC01:101:2::/48
  Known via "connected", distance 0, metric 0, type connected
  Route count is 1/1, share count 0
  Routing paths:
    directly connected via SR0
      Route metric is 0, traffic share count is 1
      Last updated 00:37:54 ago
```

Example 6: This example shows how to view the configured express forwarding path (CEF):

```
router# show ipv6 cef FC01:101:2::/48  internal
FC01:101:2::/48, epoch 0, flags [att, cnn, srsid], RIB[C], refcnt 5, per-destination sharing

  sources: SRv6-SID, RIB
  feature space:
    IPRM: 0x00038004
    Broker: linked, distributed at 2nd priority
  subblocks:
    SRv6 SID: FC01:101:2::/48
     Block-len:32 Node-len:16 Func-len:0 Arg-len:0
     END Flags:0x1 OCE:
      End OCE stats:
        packet count:  0
        byte count:    0
        punt packet count:  0
        punt byte count:    0
        error count:   0
     SRv6 end 0x80007FF32CFA6F38, 4 locks [Flags: clean]
       Lookup in input interface's IPv6 table
  ifnums: (none)
  path list 7FF32C863280, 21 locks, per-destination, flags 0x65 [shble, hvsh, rcrsv, hwcn]
    path 7FF32C85D978, share 1/1, type recursive, for IPv6
      recursive via ::[IPv6:Default], fib 7FF32C87D000, 1 terminal fib, v6:Default:::/127
      path list 7FF32C8631D0, 2 locks, per-destination, flags 0x61 [shble, rcrsv, hwcn]
          path 7FF32C85D8A8, share 1/1, type recursive, for IPv6, flags [dsnt-src-via,
cef-intnl]
           recursive via ::/127<nh:::>[IPv6:Default], fib 7FF32C3592D8, 1 terminal fib,
v6:Default:::/127
           path list 7FF32C2F5860, 5 locks, per-destination, flags 0x41 [shble, hwcn]
               path 7FF32BF8ED50, share 1/1, type special prefix, for IPv6
                 discard
  output chain:
    SRv6 end 0x80007FF32CFA6F38, 5 locks [Flags: clean]
      Lookup in input interface's IPv6 table
```

SRv6 under IS-IS

SRv6 under IS-IS

Intermediate System-to-Intermediate System (IS-IS) protocol already supports segment routing with MPLS dataplane (SR-MPLS). From Cisco IOS XE 17.12.1a, IS-IS is extended to support Segment Routing with

IPv6 data plane (SRv6). The extensions include advertising the SRv6 capabilities of nodes, node segments, and adjacency segments as SRv6 SIDs.

Information About SRv6 under IS-IS

SRv6 under IS-IS performs the following functionalities:

- Interacts with SID Manager to learn local locator prefixes and announces the locator prefixes in the IGP domain.
- Learns remote locator prefixes from other IS-IS neighbor routers and installs the learned remote locator IPv6 prefix in RIB.
- Allocates or learns prefix SID and adjacency SIDs, creates local SID entries, and advertises them in the IGP domain.

Configuring SRv6 under IS-IS

Use the **segment-routing srv6** command under the **router isis** command to enable SRv6 under the IS-IS IPv6 address-family as shown in the examples below. Use the **level** {**1**| **2**} keywords to advertise the locator only in the specified IS-IS level.

For basic SRv6 configuration, see section Configuring SRv6.

The following example shows how to configure SRv6 under IS-IS.

```
Router(config)# router isis core
Router(config-isis)# address-family ipv6 unicast
Router(config-isis-af)# router-id Loopback0
Router(config-isis-af)# segment-routing srv6
Router(config-isis-af-srv6)# locator loc5
Router(config-isis-af-srv6-locator)# level 1
Router(config-isis-srv6-locator)# exit
```

The following example shows how to assign multiple SRv6 locators under IS-IS.

```
Router(config)# router isis core
Router(config-isis)# address-family ipv6 unicast
Router(config-isis-af)# segment-routing srv6
Router(config-isis-srv6)# locator myLocBestEffort
Router(config-isis-srv6-loc)# exit
Router(config-isis-srv6)# locator myLocLowLat
Router(config-isis-srv6-loc)# exit
```

For more information about configuring IS-IS, see chapter IS-IS Overview and Basic Configuration in the *Cisco IP Routing Configuration Guide*.

Note The **router-id** keyword enables the use of SRv6 policy.

Verifying SRv6 IS-IS Configuration

Example 1: Use the **show segment-routing srv6 locator** command to verify SRv6 under IS-IS configuration:

```
Router# show segment-routing srv6 locator

Name                    ID     Algo     Prefix             Status     Flags

myLoc1                  3       0       2001:0:8::/48        Up         U
myLocBestEffort         5       0       2001:0:1::/48        Up         U
```

Example 2: Use the **show isis srv6 locators** command to view SID locators.

```
router# show isis srv6 locators
ISIS SRv6 Locators:
Tag sr:
Name              Prefix                        Level
----              ------                        -----
loc1              FC01:101:2::/48               2
```

```
router# show isis srv6 locators detail
ISIS SRv6 Locators:
Tag sr:
Name              Prefix                         Level
----              ------                         -----
loc1              FC01:101:2::/48                2

Level-1 metric: 0
 Level-2 metric: 0
 End-SIDs:
   FC01:101:2::
```

SRv6 BGP-Based Services

SRv6 BGP-Based Services

Feature Name	Release	Description
Dual-Stack L3VPN Services (IPv4, IPv6) (SRv6 Micro-SID)	Cisco IOS XE Release 17.12.1a	This feature introduces support SRV6 for VPNV4 and VPNv6 VRFs. uDT4 and uDT6 based SRv6 service on the same interface, subinterface, or VRF are supported.

Building on the messages and procedures defined in IETF draft BGP/MPLSIP Virtual Private Networks (VPNs), BGP has been extended to provide the following services over an SRv6 network:

- IPv4 Layer-3 VPNs
- IPv6 Layer-3 VPNs

Based on the messages and procedures defined in IETF draft SRv6 BGP based Overlay services, BGP encodes the SRv6 Service SID in the prefix-SID attribute of the corresponding BGP updates, and advertises it to its IPv6 BGP peers.

For more information about BGP, refer to the chapter Cisco BGP Overview in the *Cisco IP Routing Configuration Guide, Cisco IOS XE 17.x*.

Restrictions for SRv6 BGP-Based Services

- The following SRv6 BGP-based services are supported:
  - IPv4 L3VPN
  - IPv6 L3VPN
- uDT4, uDT6, and uDT46 for L3VPN are supported.
- BGP does not support uDT46 allocation and advertisement.

Information About SRv6 BGP-Based Services

SRv6 Locator Inheritance Rules

SRv6 locators can be assigned at different levels inside the BGP routing process. BGP allocates SRv6 Service SIDs from configured locator spaces according to the following inheritance rules:

1. Use the locator as defined under the service. If not defined under the specific service, then:
2. Use the locator as defined under the corresponding address-family. If not defined under the corresponding address-family, then:
3. Use the locator as defined globally under BGP.

There are multiple places under BGP where locator(s) are specified:

- Global (most generic)
- VPN AF
- VRF AF (most specific)

If there is no specific locater configured, then the locator config from upper level is inherited in the following order:

Global -> VPN-AF -> VRF-AF



Note There is no default SRv6 SID allocation mode, and Locator mode cannot be configured without SRv6 SID allocation mode. If there is no locator configured or inherited, then BGP does not allocate SIDs.

BGP Handling of SID Manager Locator Changes

In the event that BGP configured locator does not exist in the SID manager,

- BGP configuration is accepted but is not active.
- BGP generates a syslog.
- BGP listens to locator config notifications from SID manager.

In the event that BGP configured locator is created in the SID manager,

- BGP is notified by SID manager of the creation.
- BGP activates if any matching locator config.
- BGP allocates SIDs for the relevant prefixes and advertises them.

In the event that BGP configured locator is deleted from the SID manager,

- SID Manager notifies BGP of the deletion.
- BGP deactivates if any matching locator config.
- BGP deallocates SIDs for the relevant prefixes and withdraws them.

In the event that BGP configured locator is modified (i.e. locator prefix is modified) in the SID manager,

- SID Manager notifies BGP of the change.
- BGP release all SIDs associated with the previous locator prefix.
- BGP allocates new SIDs for the new locator prefix and advertise updated prefixes.

For more information on how to configure an SRv6 locator, see section ConfiguringSRv6.

SRv6 Based L3VPN

This section provides information about L3VPNs (VPNv4 and VPNv6) over an SRv6 network.

The following restrictions apply to L3VPNs over an SRv6 network:

- Only Per-VRF allocation mode is supported (uDT4 and uDT6 behavior).
- Equal-Cost Multi-path (ECMP) is supported; Unequal Cost Multipath (UCMP) is not supported.
- MPLS L3VPN and SRv6 L3VPN interworking gateway is not supported.

Configuring SRv6 based L3VPN

To enable SRv6-based L3VPN, you need to enable SRv6 under BGP, specify the locator, and configure the SID allocation mode. The assignment of the locator can be done in different places under the **router bgp** configuration. See section SRv6 Locator Inheritance Rules.

Enabling SRv6 Globally under BGP

Use the **segment-routing srv6** command under the **router bgp as-number** command to enable SRv6 globally under the BGP routing process. The *as-number* range is from 1-65535.

```
router bgp 65000
 segment-routing srv6
  locator loc1
 exit-srv6
 !
```

Configuring SRv6 IPv4 L3VPN

This example shows the complete configuration for SRv6 based IPv4 L3VPN.

```
router bgp 65000
 !
 bgp router-id interface Loopback1
 no bgp default ipv4-unicast
 neighbor 2001::1:1:1:4 remote-as 65000
 neighbor 2001::1:1:1:4 update-source Loopback1
 address-family vpnv4
  !
  segment-routing srv6
   locator loc1
   alloc-mode per-vrf
  exit-srv6
  !
  neighbor 2001::1:1:1:4 activate
  neighbor 2001::1:1:1:4 send-community both
```

Configuring SRv6 IPv6 L3VPN

This example shows the complete configuration for SRv6 based IPv6 L3VPN.

```
router bgp 65000
 !
 bgp router-id interface Loopback1
 no bgp default ipv4-unicast
 neighbor 2001::1:1:1:4 remote-as 65000
 neighbor 2001::1:1:1:4 update-source Loopback1
 address-family vpnv6
  !
  segment-routing srv6
   locator loc1
   alloc-mode per-vrf
  exit-srv6
  !
  neighbor 2001::1:1:1:4 activate
  neighbor 2001::1:1:1:4 send-community both
```

Configuring SRv6 IPvx VRF L3VPN

This example shows the complete configuration for SRv6 based L3VPN for address family IPvx VRF.

```
router bgp 65000
 !
 bgp router-id interface Loopback1
 no bgp default ipv4-unicast
 neighbor 2001::1:1:1:4 remote-as 65000
 neighbor 2001::1:1:1:4 update-source Loopback1
 address-family ipv4 vrf ce1
  !
  segment-routing srv6
   locator loc1
   alloc-mode per-vrf
  exit-srv6
  !
  neighbor 99.1.2.1 remote-as 65001
  neighbor 99.1.2.1 activate
  neighbor 99.1.2.1 send-community both
 address-family ipv6 vrf ce1
  !
  segment-routing srv6
   locator loc1
   alloc-mode per-vrf
  exit-srv6
```

```
!
neighbor 1002::1 remote-as 65002
neighbor 1002::1 activate
neighbor 1002::1 send-community both
```

SRv6 eBGP Inter-AS

From Cisco IOS XE 17.16.1a, the following BGP inter-autonomous systems (Inter-AS) functionalities are supported over SRv6:

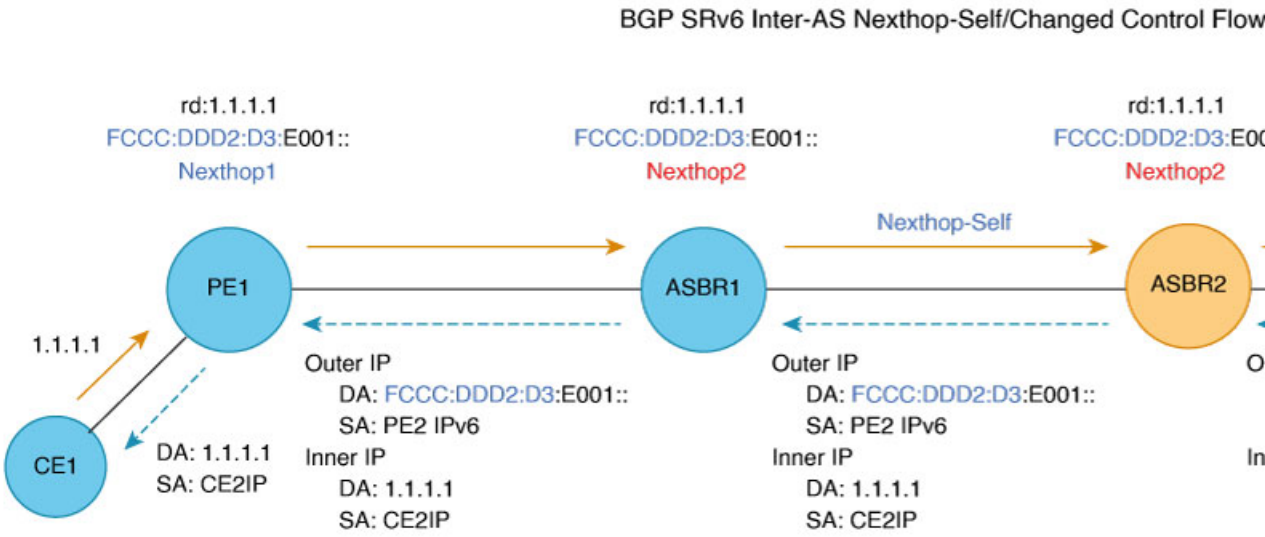
- BGP SRv6 Inter-AS Nexthop-Self/Changed
- BGP SRv6 Inter-AS Nexthop Unchanged

eBGP SRv6 Inter-AS Nexthop-Self/Changed

In this scenario, autonomous system boundary routers (ASBRs) propagate the BGP SRv6 L3VPN updates with SRv6 SID unchanged, but the next-hop destination changed. The reachability of the Egress PE to the SRv6 locator must be advertised across AS domains.

End-to-End Inter-AS SRv6 SRTE policy cannot be created by the headend node; it can only create SRTE policy for the ASBRs.

Figure 41: eBGP SRv6 Inter-AS Nexthop-Self/Changed

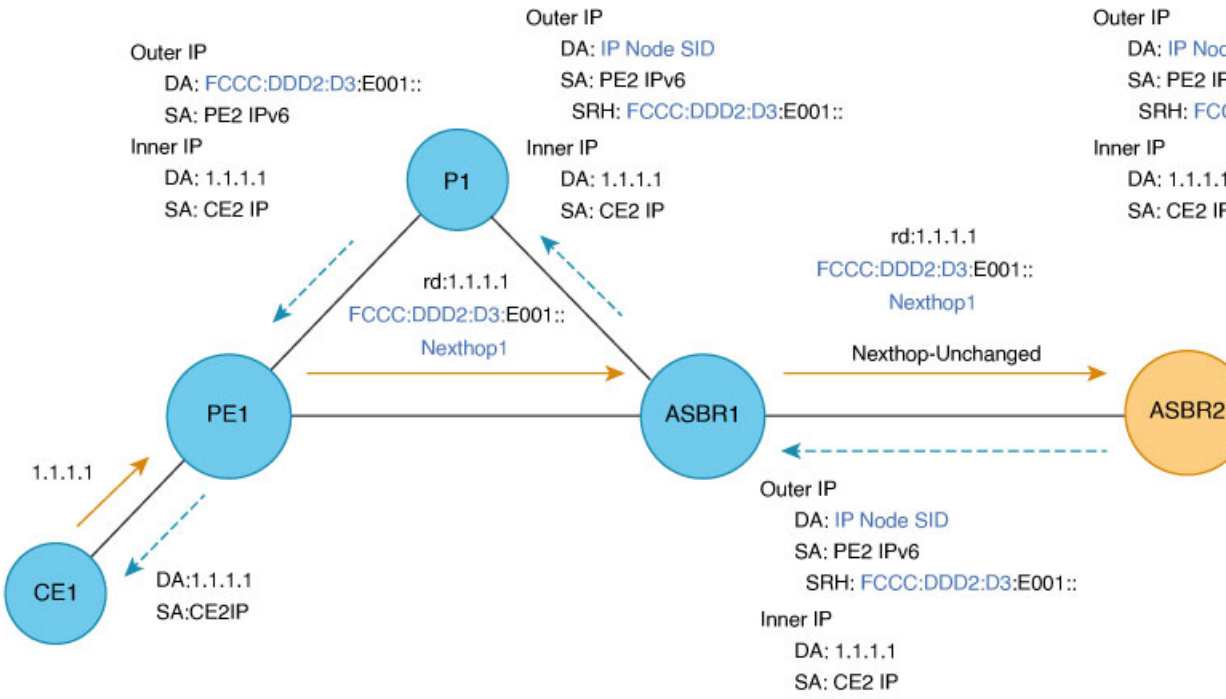


BGP SRv6 Inter-AS Nexthop Unchanged

In this scenario, ASBRs propagate the BGP SRv6 L3VPN updates with both SRv6 SID and next-hop unchanged. The reachability of the next-hop of the Egress PE to the SRv6 locator must be advertised across AS domains.

End-to-End Inter-AS SRv6 SRTE policy can be created by the headend node to the ingress endpoint if the complete network is managed by single IGP domain.

Figure 42: eBGP SRv6 Inter-AS Nexthop Unchanged



Configure SRv6 eBGP Inter-AS Nexthop Unchanged

The following sample configuration shows how to configure eBGP inter-AS with the SRv6 SID as well as the nexthop destination unchanged:

```
router bgp 100
  neighbor A001::1 disable-connected-check
  neighbor A001::1 remote-as 200
  neighbor A001::1 ebgp-multihop 255
  neighbor A001::1 update-source Loopback20
!
 address-family vpnv4
  neighbor A001::1 encap srv6
  neighbor A001::1 activate
  neighbor A001::1 send-community both
  neighbor A001::1 next-hop-unchanged
!
 address-family vpnv6
  neighbor A001::1 encap srv6
  neighbor A001::1 activate
  neighbor A001::1 send-community both
  neighbor A001::1 next-hop-unchanged
```

Configure SRv6 eBGP Inter-AS Nexthop Unchanged

The following sample configuration shows how to configure eBGP inter-AS with the SRv6 SID as well as the nexthop destination unchanged:

```
router bgp 100
  neighbor A001::1 disable-connected-check
  neighbor A001::1 remote-as 200
```

```
  neighbor A001::1 ebgp-multihop 255
  neighbor A001::1 update-source Loopback20
!
 address-family vpnv4
  neighbor A001::1 encap srv6
  neighbor A001::1 activate
  neighbor A001::1 send-community both
  neighbor A001::1 next-hop-unchanged
!
 address-family vpnv6
  neighbor A001::1 encap srv6
  neighbor A001::1 activate
  neighbor A001::1 send-community both
  neighbor A001::1 next-hop-unchanged
```

Verify SRv6 eBGP Inter-AS

Use the **show run** command to view the ASBR next-hop in an SRv6 eBGP configuration:

```
router#show run | se router bgp 100
router bgp 100
 neighbor A001::1 encap srv6
 neighbor A001::1 remote-as 200
 neighbor A001::1 ebgp-multihop 255
 neighbor A001::1 update-source Loopback20
!
 address-family vpnv4
  neighbor A001::1 encap srv6
  neighbor A001::1 activate
  neighbor A001::1 send-community both
  neighbor A001::1 next-hop-unchanged
```

```
router#show run | se router bgp 100
router bgp 100
 neighbor A001::1 encap srv6
 neighbor A001::1 remote-as 200
 neighbor A001::1 ebgp-multihop 255
 neighbor A001::1 update-source Loopback20
!
 address-family vpnv4
  neighbor A001::1 encap srv6
  neighbor A001::1 activate
  neighbor A001::1 send-community both
  neighbor A001::1 next-hop-self
```

BGP MPLS and SRv6 Co-Existence

A dual-connected PE that has both MPLS and SRv6 neighbors concurrently allocates a local MPLS label and an SRv6 SID for sourced/CE routes.

Restrictions

- MPLS label allocation is disabled when SRV6 is enabled for BGP AFI VRF.
- The **mpls alloc enable** command enables MPLS label allocation and is the default allocation mode. Both SRV6 and MPLS allocations are enabled, with MPLS being default allocation mode.
- MPLS label is advertised to a neighbor by default in the MPLS and SRv6 co-existence configuration.
- The **neighbor <> encap srv6** command is required to advertise SRv6 SID to a neighbor.

Configuring MPLS and SRv6 Coexistence for L3VPN

The following example shows the configuration to enable MPLS and SRv6 co-existence for L3VPN:

```
router bgp <instance>
    address-family [ipv4 | ipv6] unicast vrf <vrf-name>
        segment-routing srv6
                mpls alloc enable           >>>>> required for MPLS/SRv6 coexistence
    address-family vpnv4/vpnv6
                neighbor <A>                >>>>> can send any kind of update
                neighbor <B> encap srv6     >>>> SRv6 only neighbor
```



Note Sourced or CE prefixes from VRF's with MPLS and SRv6 coexistence enabled will be sent with MPLS labels.

Verifying SRv6 State

Use the following show commands to verify SRv6 BGP configurations.

Example 1: **`show segment-routing srv6 sid`**

```
device# show segment-routing srv6 sid
SID                   Locator     Behavior          Context                          Owner
---                   -------     --------          -------                          -----
FC01:101:2::          loc1        uN (PSP/USD)                                       SID-MGR

FC01:101:2:E000::     loc1        uDT4              ce1                              router
 bgp
FC01:101:2:E001::     loc1        uDT6              ce1                              router
 bgp
FC01:101:2:E002::     loc1        uA (PSP/USD)      Ethernet2/0 2001::99:2:3:3       router
 isis sr
FC01:101:2:E003::     loc1        uA (PSP/USD)      Ethernet2/1 2001::100:2:3:3      router
 isis sr
FC01:101:2:E004::     loc1        uA (PSP/USD)      Ethernet3/0 2001::99:2:4:4       router
 isis sr
FC01:101:2:E005::     loc1        uA (PSP/USD)      Ethernet3/1 2001::100:2:4:4      router
 isis sr
FC01:101:2:E006::     loc1        uA (PSP/USD)      Ethernet4/0 2001::99:2:5:5       router
 isis sr
FC01:101:2:E007::     loc1        uA (PSP/USD)      Ethernet4/1 2001::100:2:5:5      router
 isis sr
```

Example 2: **`show segment-routing srv6 sid <SID> detail`**

```
device# show segment-routing srv6 sid FC01:101:2:E000:: detail
SID: FC01:101:2:E000::   Type: DYNAMIC
 Behavior: uDT4 (63)
 Context:
    interface: (not-set)
    vrf: ce1, v4-topo-id: 0x1, v6-topo-id: 0xFFFF
    next-hop: (not-set)
    policy: (not-set)
    distinguisher: (not-set)
 Stats:
    Packets: 0  Bytes: 0
 User list:
    User:Refcount        Locator:Refcount
```

```
    -------------        -----------------
    router bgp(5):1      loc1:1
 Event history:
    Timestamp            Client           Event type
    ---------            ------           ----------
    04-15 07:24:08.165   router bgp(5)    ALLOC
```

Example 3: `show ip bgp all srv6 locator`

```
device# show ip bgp all srv6 locator
 Locator-1
   Name:  loc1
   Active:  Yes
   Refcount:  3
```

Example 4: `show ip bgp all srv6 sid`

```
device# show ip bgp all srv6 sid
 SID-1
   locator : loc1
   alloc-mode : 0
   status : ALLOCATED
   state : 1
   ref_count : 5
   topoid : 0x1E000001
   sid_value : FC01:101:2:E001::
   prefix_length : 64
   block_length : 32
   node _length : 16
   function _length : 16
   arg_length : 0
   behaviour : 62
 SID-2
   locator : loc1
   alloc-mode : 0
   status : ALLOCATED
   state : 1
   ref_count : 5
   topoid : 0x1
   sid_value : FC01:101:2:E000::
   prefix_length : 64
   block_length : 32
   node _length : 16
   function _length : 16
   arg_length : 0
   behaviour : 63
```

Example 5: `show ipv6 cef <prefix> internal`

```
device# show ipv6 cef FC01:101:8:E006:: internal
FC01:101:8:E006::/128, epoch 0, flags [att, srsid], refcnt 4, per-destination sharing
  sources: SRv6-SID
  subblocks:
    SRv6 SID: FC01:101:8:E006::/128
     Block-len:32 Node-len:16 Func-len:16 Arg-len:0
     END-DT4 Flags:0x5 OCE:
      End OCE stats:
        packet count:  20
        byte count:    2280
        punt packet count:  0
        punt byte count:    0
        error count:   0
```

```
     SRv6 end 0x80007FD05D9BC970, 4 locks [Flags: clean decap]
       Lookup in table IPv4:ce2
  ifnums: (none)
  path list 7FD05BD3F530, 21 locks, per-destination, flags 0x65 [shble, hvsh, rcrsv, hwcn]
    path 7FD05BD2D578, share 1/1, type recursive, for IPv6
      recursive via ::[IPv6:Default], fib 7FD05BD43C60, 1 terminal fib, v6:Default:::/127
      path list 7FD05BD3F480, 2 locks, per-destination, flags 0x61 [shble, rcrsv, hwcn]
          path 7FD05BD2D4A8, share 1/1, type recursive, for IPv6, flags [dsnt-src-via,
cef-intnl]
            recursive via ::/127<nh:::>[IPv6:Default], fib 7FD056DAB760, 1 terminal fib,
v6:Default:::/127
            path list 7FD054328EF8, 5 locks, per-destination, flags 0x41 [shble, hwcn]
                path 7FD05AF52578, share 1/1, type special prefix, for IPv6
                  discard
  output chain:
    SRv6 end 0x80007FD05D9BC970, 5 locks [Flags: clean decap]
      Lookup in table IPv4:ce2
```

Example 6: `show isis database verbose`

```
device# show isis database verbose
pe3.00-00          0x00000025   0xEF58                742/1198      0/0/0
  Area Address: 49
  NLPID:        0xCC 0x8E
  Topology:     IPv4 (0x0)
                IPv6 (0x2)
  Router ID:    1.1.1.8
  Router CAP:   1.1.1.8, D:0, S:0
    SRv6 Oflag:0
    Segment Routing: I:1 V:0, SRGB Base: 16000 Range: 8000
    Segment Routing Algorithms: SPF, Strict-SPF
    Segment Routing Local Block: SRLB Base: 30000 Range: 10000
    Node-MSD
      MSD: 16
  Hostname: iolpe3
(snip)
 SRv6 Locator: (MT-IPv6) FC01:101:8::/48 Metric:0 Algorithm:0
    End SID: FC01:101:8:: uN (PSP/USD)
      SID Structure:
        Block Length: 32, Node-ID Length: 16, Func-Length: 0, Args-Length: 0
```

Example 7: `show ipv6 route <prefix>`

```
device# show ipv6 route FC01:101:8::/48
Routing entry for FC01:101:8::/48
  Known via "isis sr", distance 115, metric 30, type level-2
  Route count is 4/4, share count 0
  Routing paths:
    FE80::A8BB:CCFF:FE01:E411, Ethernet3/1
      Route metric is 30, traffic share count is 1
      From FE80::A8BB:CCFF:FE01:E411
      Last updated 01:03:27 ago
    FE80::A8BB:CCFF:FE03:F504, Ethernet4/0
      Route metric is 30, traffic share count is 1
      From FE80::A8BB:CCFF:FE03:F504
      Last updated 01:03:27 ago
    FE80::A8BB:CCFF:FE03:F514, Ethernet4/1
      Route metric is 30, traffic share count is 1
      From FE80::A8BB:CCFF:FE03:F514
      Last updated 01:03:27 ago
    FE80::A8BB:CCFF:FE01:E401, Ethernet3/0
      Route metric is 30, traffic share count is 1
```

```
      From FE80::A8BB:CCFF:FE01:E401
      Last updated 01:03:27 ago
```

Example 8: **`show bgp [vpnv4|vpnv6] rd <rd> <prefix>`**

Sample output for VPNv4:

```
device# show bgp vpnv4 uni rd 1:1 22.22.22.22

BGP routing table entry for 1:1:22.22.22.22/32, version 13
Paths: (1 available, best #1, table red)
Not advertised to any peer
Refresh Epoch 1
3, imported path from 2:2:22.22.22.22/32 (global)
2023:1::1 (via default) from 1.1.1.3 (1.1.1.3)
Origin incomplete, metric 0, localpref 100, valid, internal, best
Extended Community: RT:1:1 RT:2:2
Originator: 11.1.1.1, Cluster list: 1.1.1.3
srv6 out-sid: FCCC:CCC1:AA88:E000::
rx pathid: 0, tx pathid: 0x0
Updated on Jun 28 2023 11:29:52 PST
```

Sample output for VPNv6:

```
device# show bgp vpnv6 uni rd 1:1 2222::1/128

BGP routing table entry for [1:1]2222::1/128, version 11
Paths: (1 available, best #1, table red)
Not advertised to any peer
Refresh Epoch 1
3, imported path from [2:2]2222::1/128 (global)
2023:1::1 (via default) from 1.1.1.3 (1.1.1.3)
Origin incomplete, metric 0, localpref 100, valid, internal, best
Extended Community: RT:2:2
Originator: 11.1.1.1, Cluster list: 1.1.1.3
srv6 out-sid: FCCC:CCC1:AA88:E001::
rx pathid: 0, tx pathid: 0x0
Updated on Jun 28 2023 11:29:52 PST
```

Example 9: **`show ip route vrf <vrf> <prefix>`**

```
device# show ip route vrf ce1 1.1.1.10
Routing Table: ce1
Routing entry for 1.1.1.10/32
  Known via "bgp 65000", distance 200, metric 0
  Tag 65010, type internal
  Last update from FC01:101:8:E006:: 08:51:34 ago
  Routing Descriptor Blocks:
  * FC01:101:8:E006:: (default:ipv6), from 1.1.1.4, 08:51:34 ago
      opaque_ptr 0x7FF32E0B9640
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 65010
      MPLS label: none
```

Example 10: **`show ipv6 route vrf <vrf> <prefix>`**

```
device# show ipv6 route vrf red 2222::1/128

Routing entry for 2222::1/128
Known via "bgp 1", distance 200, metric 0
Tag 3, type internal
```

```
Route count is 1/1, share count 0
Routing paths:
FCCC:CCC1:AA88:E001::%default
Route metric is 0, traffic share count is 1
From ::FFFF:1.1.1.3
opaque_ptr 0x7FF38CDB6848
Last updated 00:03:16 ago
```

Example 11: **`show ip cef vrf <vrf> <prefix> internal`**

```
device# show ip cef vrf red 22.22.22.22 internal

22.22.22.22/32, epoch 0, flags [rnolbl, rlbls], RIB[B], refcnt 5, per-destination sharing
  sources: RIB
  feature space:
    IPRM: 0x00018000
  VPN-SID(s) on: 1/0:v4-rcrsv-FCCC:CCC1:AA88:E000::
  Path: v4-rcrsv-FCCC:CCC1:AA88:E000:: (VPN-SID: FCCC:CCC1:AA88:E000::)
    Flags: 00000004 [vpn-sid]
    IPv6 TC: 0   Hop Limit: 64
      Src: C02:1::7
      Dst: FCCC:CCC1:AA88:E000::
      Via: FCCC:CCC1:AA88:E000::
    Segment List (1)
      FCCC:CCC1:AA88:E000::
    Flow-based Encap Chains: 1
      IPV6 adj out of Ethernet0/0, addr FE80::A8BB:CCFF:FE00:3300 from FCCC:CCC1:AA88::/48
 <= SRv6 SID List OCE 0x7FF38D329078 (5) 1 Segments
  ifnums:
    Ethernet0/0(2): FE80::A8BB:CCFF:FE00:3300
  path list 7FF38CCDE0D8, 7 locks, per-destination, flags 0x8269 [shble, rif, rcrsv, hwcn,
 bgp, sb-oce]
    path 7FF38CCDB128, share 1/1, type recursive, for IPv4, flags [vpn-sid],
vpn-sid:FCCC:CCC1:AA88:E000::
      recursive via FCCC:CCC1:AA88:E000::[IPv6:Default], fib 7FF38CDA31B0, 1 terminal fib,
 v6:Default:FCCC:CCC1:AA88::/48
     path list 7FF38CCDDE18, 2 locks, per-destination, flags 0x69 [shble, rif, rcrsv, hwcn]

          path 7FF38CCDADE8, share 1/1, type recursive, for IPv6, flags [dsnt-src-via,
cef-intnl]
           recursive via FCCC:CCC1:AA88::/48<nh:FCCC:CCC1:AA88:E000::>[IPv6:Default], fib
 7FF38CDA3D78, 1 terminal fib, v6:Default:FCCC:CCC1:AA88::/48
           path list 7FF38CCDE658, 5 locks, per-destination, flags 0x49 [shble, rif, hwcn]

               path 7FF38CCDB7A8, share 1/1, type attached nexthop, for IPv6
               nexthop FE80::A8BB:CCFF:FE00:3300 Ethernet0/0, IPV6 adj out of Ethernet0/0,
 addr FE80::A8BB:CCFF:FE00:3300 7FF38CDE1848
  output chain:
    SRv6 SID List OCE 0x7FF38D329078 (8) 1 Segments
      Segment List (1)
        FCCC:CCC1:AA88:E000::
    PushCounter(SRv6 Encap) 7FF386CF0E58
    SRv6 Encap OCE 0x7FF38D328BE8 (4) fwd-id:0 FCCC:CCC1:AA88:E000::
      Flags: 00000004 [vpn-sid]
      IPv6 TC: 0   Hop Limit: 64
        Src: C02:1::7
        Dst: FCCC:CCC1:AA88:E000::
    IPV6 adj out of Ethernet0/0, addr FE80::A8BB:CCFF:FE00:3300 7FF38CDE1848
```

Example 12: **`show ipv6 cef vrf <vrf> <prefix> internal`**

```
device# show ipv6 cef vrf red 2222::1/128 internal

2222::1/128, epoch 0, RIB[B], refcnt 4, per-destination sharing
  sources: RIB
  feature space:
    IPRM: 0x00018000
  VPN-SID(s) on: 1/0:v6-rcrsv-FCCC:CCC1:AA88:E001::
  Path: v6-rcrsv-FCCC:CCC1:AA88:E001:: (VPN-SID: FCCC:CCC1:AA88:E001::)
    Flags: 00000004 [vpn-sid]
    IPv6 TC: 0   Hop Limit: 64
      Src: C02:1::7
      Dst: FCCC:CCC1:AA88:E001::
      Via: FCCC:CCC1:AA88:E001::
    Segment List (1)
      FCCC:CCC1:AA88:E001::
    Flow-based Encap Chains: 1
      IPV6 adj out of Ethernet0/0, addr FE80::A8BB:CCFF:FE00:3300 from FCCC:CCC1:AA88::/48
 <= SRv6 SID List OCE 0x7FF38D329018 (6) 1 Segments
  ifnums:
    Ethernet0/0(2): FE80::A8BB:CCFF:FE00:3300
  path list 7FF38CCDDD68, 9 locks, per-destination, flags 0x8269 [shble, rif, rcrsv, hwcn,
 bgp, sb-oce]
    path 7FF38CCDAD18, share 1/1, type recursive, for IPv6, flags [vpn-sid],
vpn-sid:FCCC:CCC1:AA88:E001::
      recursive via FCCC:CCC1:AA88:E001::[IPv6:Default], fib 7FF38CDA2E10, 1 terminal fib,
 v6:Default:FCCC:CCC1:AA88::/48
     path list 7FF38CCDDCB8, 2 locks, per-destination, flags 0x69 [shble, rif, rcrsv, hwcn]

          path 7FF38CCDAC48, share 1/1, type recursive, for IPv6, flags [dsnt-src-via,
cef-intnl]
           recursive via FCCC:CCC1:AA88::/48<nh:FCCC:CCC1:AA88:E001::>[IPv6:Default], fib
 7FF38CDA3D78, 1 terminal fib, v6:Default:FCCC:CCC1:AA88::/48
           path list 7FF38CCDE658, 5 locks, per-destination, flags 0x49 [shble, rif, hwcn]

                path 7FF38CCDB7A8, share 1/1, type attached nexthop, for IPv6
                nexthop FE80::A8BB:CCFF:FE00:3300 Ethernet0/0, IPV6 adj out of Ethernet0/0,
 addr FE80::A8BB:CCFF:FE00:3300 7FF38CDE1848
  output chain:
    SRv6 SID List OCE 0x7FF38D329018 (9) 1 Segments
      Segment List (1)
        FCCC:CCC1:AA88:E001::
    PushCounter(SRv6 Encap) 7FF386CF0DC8
    SRv6 Encap OCE 0x7FF38D328B48 (4) fwd-id:0 FCCC:CCC1:AA88:E001::
      Flags: 00000004 [vpn-sid]
      IPv6 TC: 0   Hop Limit: 64
        Src: C02:1::7
        Dst: FCCC:CCC1:AA88:E001::
    IPV6 adj out of Ethernet0/0, addr FE80::A8BB:CCFF:FE00:3300 7FF38CDE1848
device#
```

Troubleshooting and Debugging SRv6 BGP

The following BGP commands can be used to debug BGP updates:

- **debug bgp <> updates**
- **debug bgp <> addpath**

The following new command is introduced for debugging events related to SRv6:

- **debug ip bgp srv6**

EVPN VPWS over SRv6

Feature history for EVPN VPWS over SRv6

Table 50: Feature history

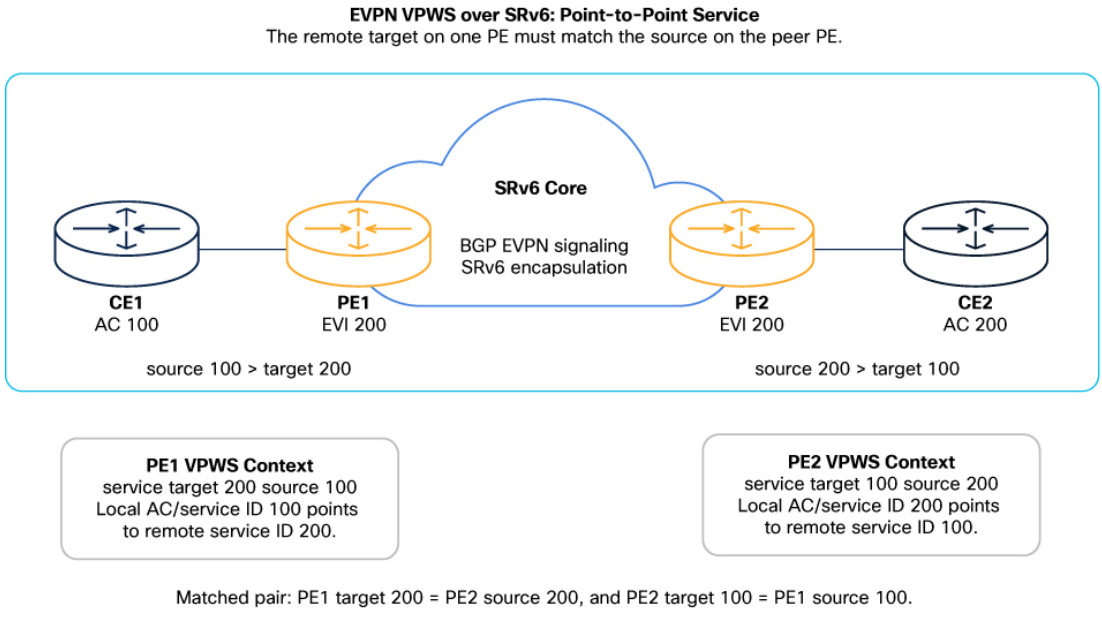
Feature Name	Release	Feature Information
EVPN VPWS over SRv6	Cisco IOS XE 26.2.1	This feature adds support for EVPN VPWS over SRv6 transport. EVPN VPWS uses EVPN signalling and SRv6 encapsulation to provide point-to-point Layer 2 VPN service between provider edge devices.

EVPN VPWS over SRv6

EVPN VPWS over SRv6 provides point-to-point Layer 2 VPN service using EVPN signalling and SRv6 encapsulation.

- EVPN VPWS forwards customer traffic to and from a customer attachment circuit without MAC lookup.
- EVPN VPWS establishes an EVPN instance between a pair of provider edge devices.
- The service uses BGP EVPN signalling and SRv6 encapsulation to carry VPWS traffic across the SRv6 transport.

Figure 43: EVPN VPWS over SRv6 point-to-point service



Supported platforms

- Cisco ASR1000
- Cisco Catalyst 8000 series edge
- Cisco Catalyst 8000V Edge Software
- Cisco 8000 Series Secure Routers

Restrictions for EVPN VPWS over SRv6

- Multihoming is not supported.
- Ethernet OAM interworking with EVPN VPWS over SRv6 is not upported.
- Remote link failure notification via EVPN VPWS over SRv6 is not supported on these platforms:
  - Cisco Catalyst 8500 Series Edge Platforms: C8500L-8S4X
  - Cisco Catalyst 8300 Series Edge Platforms: C8300-2N2S-4T2X
  - Cisco 1000 Series Integrated Services Routers: C1131X
  - 8100 Series Secure Routers: C8130-G2
  - 8200 Series Secure Routers
  - 8300 Series Secure Routers
  - 8400 Series Secure Routers: C8475-G2

Configure EVPN VPWS over SRv6

Perform the following steps to configure EVPN VPWS over SRv6:

Procedure

Step 1 Configure SRv6 with a locator.

```
segment-routing srv6
locators
  locator locator1
   prefix 2001:1:1::/48
   format usid-f3216
```

Step 2 Configure IGP with SRv6.

Configure IS-IS (or another IGP) to advertise SRv6 locator prefixes for IPv6 reachability.

Note: Multiple locators can be assigned to the IGP for advertisement.

```
router isis isis-1
 net 49.0001.0000.0000.0001.00
 is-type level-2-only
 router-id Loopback0
 metric-style wide
```

```
 !
 address-family ipv6
  segment-routing srv6
   locator locator2
    level-2
exit-address-family
```

Step 3 Configure BGP for EVPN-VPWS over SRv6.

Configure BGP with the L2VPN EVPN address family and enable SRv6 encapsulation for the EVPN neighbor. For eBGP EVPN neighbors, neighbor <address> encap srv6 is required to advertise EVPN SRv6 updates. For iBGP EVPN neighbors, this command is optional.

```
router bgp 1001
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 2001:101:4:4::4 remote-as 1001
 neighbor 2001:101:4:4::4 update-source Loopback0
 !
 address-family l2vpn evpn
  neighbor 2001:101:4:4::4 activate
  neighbor 2001:101:4:4::4 send-community both
  neighbor 2001:101:4:4::4 encap srv6
 exit-address-family
```

Step 4 Configuring the EVPN VPWS Global SRv6 Locator

The global SRv6 locator under l2vpn evpn is used as the default locator for EVPN VPWS instances that use SRv6 encapsulation, unless a more specific locator is configured at the instance or VPWS context level.

```
l2vpn evpn
 logging vpws vc-state
 router-id Loopback0
 segment-routing srv6 locator locator1
```

Step 5 Configuring the EVPN-VPWS Instance with SRv6 Encapsulation

The encapsulation srv6 command enables SRv6 transport for the EVPN VPWS instance. If encapsulation is not configured, MPLS is used as the default transport.

The SRv6 locator can be configured globally under l2vpn evpn, at the EVPN instance level, or under the VPWS context. The most specific locator takes precedence.

```
l2vpn evpn instance 1 point-to-point
 encapsulation srv6
 segment-routing srv6 locator locator1 (optional)
 vpws context vc_1
  service target 20000 source 10000
  member TenGigabitEthernet0/0/2 service-instance 1
  segment-routing srv6 locator locator1 (optional)
 !
 vpws context vc_2
  service target 20001 source 10001
  member TenGigabitEthernet0/0/2 service-instance 10
  segment-routing srv6 locator locator1 (optional)
 !
```

Step 6 Configuring the CE-Facing Interface

EVPN VPWS over SRv6 supports the following attachment circuit interface modes:

- Port mode, using the main interface

```
 ! Port mode / main interface
l2vpn evpn instance 200 point-to-point
 encapsulation srv6
 vpws context vc200
  service target 200 source 200
  member TenGigabitEthernet0/0/8
```

- Sub-interface mode

```
 ! Sub-interface mode
interface TenGigabitEthernet0/0/8.200
 encapsulation dot1q 200
!
l2vpn evpn instance 200 point-to-point
 encapsulation srv6
 vpws context vc200
  service target 200 source 200
  member TenGigabitEthernet0/0/8.200
```

- Ethernet Flow Point (EFP) mode, using an Ethernet service instance

```
 ! EFP mode
interface TenGigabitEthernet0/0/8
 service instance 200 ethernet
  encapsulation dot1q 200
!
l2vpn evpn instance 200 point-to-point
 encapsulation srv6
 vpws context vc200
  service target 200 source 200
  member TenGigabitEthernet0/0/8 service-instance 200
```

EVPN VPWS over SRv6 configuration examples

The following examples show running configurations on PE1, PE2 and RR in an EVPN-VPWS over SRv6 deployment.

Example: PE1 Configuration

```
! Loopback Interface
interface Loopback0
 ipv6 address 2001:101:1:1::1/128
 ipv6 router isis isis-1
! Core-Facing Interfaces
interface TenGigabitEthernet0/0/0
 ipv6 address 2001:101:2:1::1/64
 ipv6 router isis isis-1
 isis network point-to-point
! CE-Facing Interface
interface TenGigabitEthernet0/0/2
 no ip address
 service instance 1 ethernet
  encapsulation dot1q 101
 !
 service instance 10 ethernet
```

```
  encapsulation default
 !
! SRv6 Configuration
segment-routing srv6
 locators
  locator locator1
   prefix 2001:1:1::/48
   format usid-f3216
! IGP Configuration
router isis isis-1
 net 49.0001.0000.0000.0001.00
 is-type level-2-only
 router-id Loopback0
 metric-style wide
 !
 address-family ipv6
  segment-routing srv6
   locator locator1
    level-2
 exit-address-family
! BGP Configuration
router bgp 1001
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 2001:101:4:4::4 remote-as 1001
 neighbor 2001:101:4:4::4 update-source Loopback0
 !
 address-family l2vpn evpn
  neighbor 2001:101:4:4::4 activate
  neighbor 2001:101:4:4::4 send-community both
  neighbor 2001:101:4:4::4 encap srv6
 exit-address-family
! EVPN VPWS Configuration
l2vpn evpn
 logging vpws vc-state
 router-id Loopback0
 segment-routing srv6 locator locator1
l2vpn evpn instance 1 point-to-point
 encapsulation srv6
 vpws context vc_1
  service target 20000 source 10000
  member TenGigabitEthernet0/0/2 service-instance 1
 !
 vpws context vc_2
  service target 20001 source 10001
  member TenGigabitEthernet0/0/2 service-instance 10
 !
```

Example: PE2 Configuration

PE2 configuration is symmetric to PE1.

The service target and source identifiers are reversed (PE2's source matches PE1's target and vice versa).

The SRv6 locator prefix and loopback addresses are unique to PE2.

The BGP neighbor address points to PE1 (or the route reflector).

```
! Loopback Interface
interface Loopback0
 ipv6 address 2001:101:6:6::6/128
 ipv6 router isis isis-1
! Core-Facing Interfaces
```

```
interface TenGigabitEthernet0/0/0
 ipv6 address 2001:101:5:6::6/64
 ipv6 router isis isis-1
 isis network point-to-point
! CE-Facing Interface
interface TenGigabitEthernet0/0/5
 no ip address
 service instance 1 ethernet
  encapsulation dot1q 101
 !
 service instance 10 ethernet
  encapsulation default
 !
! SRv6 Configuration
segment-routing srv6
 locators
  locator locator1
   prefix 2001:1:6::/48
   format usid-f3216
! IGP Configuration
router isis isis-1
 net 49.0001.0000.0000.0006.00
 is-type level-2-only
 router-id Loopback0
 metric-style wide
 !
 address-family ipv6
  segment-routing srv6
   locator locator1
    level-2
 exit-address-family
! EVPN VPWS Configuration
l2vpn evpn
 logging vpws vc-state
 router-id Loopback0
 segment-routing srv6 locator locator1
l2vpn evpn instance 1 point-to-point
 encapsulation srv6
 vpws context vc_1
  service target 10000 source 20000
  member TenGigabitEthernet0/0/5 service-instance 1
 !
 vpws context vc_2
  service target 10001 source 20001
  member TenGigabitEthernet0/0/5 service-instance 10
 !
! BGP Configuration
router bgp 1001
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 2001:101:4:4::4 remote-as 1001
 neighbor 2001:101:4:4::4 update-source Loopback0
 !
 address-family l2vpn evpn
  neighbor 2001:101:4:4::4 activate
  neighbor 2001:101:4:4::4 send-community both
  neighbor 2001:101:4:4::4 encap srv6
 exit-address-family
Important: Ensure the service identifiers are symmetric across PEs. PE1's source 10000 /
target 20000 must match PE2's source 20000 / target 10000.
```

Example: RR Configuration

```
interface Loopback0
 ipv6 address 2001:101:4:4::4/128
 ipv6 router isis isis-1
! SRv6 Configuration
segment-routing srv6
 locators
  locator locator1
   prefix 2001:188:4::/48
   format usid-f3216
! IGP Configuration
router isis isis-1
 net 49.0001.0000.0000.0004.00
 is-type level-2-only
 router-id Loopback0
 metric-style wide
 !
 address-family ipv6
  segment-routing srv6
   locator locator1
    level-2
 exit-address-family
! BGP Configuration
router bgp 1001
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 2001:101:1:1::1 remote-as 1001
 neighbor 2001:101:1:1::1 log-neighbor-changes
 neighbor 2001:101:1:1::1 update-source Loopback0
 neighbor 2001:101:6:6::6 remote-as 1001
 neighbor 2001:101:6:6::6 log-neighbor-changes
 neighbor 2001:101:6:6::6 update-source Loopback0
 !
address-family l2vpn evpn
  neighbor 2001:101:1:1::1 activate
  neighbor 2001:101:1:1::1 send-community both
  neighbor 2001:101:1:1::1 route-reflector-client
  neighbor 2001:101:1:1::1 encap srv6
  neighbor 2001:101:6:6::6 activate
  neighbor 2001:101:6:6::6 send-community both
  neighbor 2001:101:6:6::6 route-reflector-client
  neighbor 2001:101:6:6::6 encap srv6
 exit-address-family
```

Verify EVPN VPWS over SRv6

Verifying SRv6 Locators

Use the show segment-routing srv6 locator command to verify that the configured SRv6 locators are in the Up state.

```
Device# show segment-routing srv6 locator
Name            Algo Prefix                                 Format              Status
----            ---- ------                                 ------              ------
locator1        0    2001:1:1::/48                          usid-f3216          Up
```

Verifying SRv6 SIDs

Use the show segment-routing srv6 sid command to verify SRv6 SID allocations. Look for uDX2 entries owned by evpn-srv6, which represent the local EVPN VPWS SIDs.

```
Device# show segment-routing srv6 sid
SID                                 Locator         Behavior
Context                                 Owner
---                                 -------         --------
-------                                 -----
2001:1:1::                          locator1        uN
(PSP/USD)                                                   SID-MGR
2001:1:1:E000::                   locator1       uDX2 (shift)         evi:1 eth_tag:10000
 if:Te0/0/2 efp:1     evpn-srv6
2001:1:1:E001::                   locator1       uDX2 (shift)         evi:1 eth_tag:10001
 if:Te0/0/2 efp:10    evpn-srv6
```

The uDX2 behavior indicates the Endpoint with decapsulation and Layer 2 cross-connect function. The context shows the EVI, Ethernet tag, interface, and EFP associated with each SID.

Verifying BGP EVPN Routes

Use the show bgp l2vpn evpn command to view the EVPN routing table summary.

```
Device# show bgp l2vpn evpn
BGP table version is 7, local router ID is 101.1.1.1
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 101.1.1.1:1
*>   [1][101.1.1.1:1][00000000000000000000][10000]/23
                      ::                                    32768 ?
*>   [1][101.1.1.1:1][00000000000000000000][10001]/23
                      ::                                    32768 ?
*>i  [1][101.1.1.1:1][00000000000000000000][20000]/23
                      2001:101:6:6::6          0    100      0 ?
*>i  [1][101.1.1.1:1][00000000000000000000][20001]/23
                      2001:101:6:6::6          0    100      0 ?
```

Use the show bgp l2vpn evpn detail command to view detailed route information including SRv6 SIDs.

```
Device# show bgp l2vpn evpn detail
Route Distinguisher: 101.1.1.1:1
BGP routing table entry for [1][101.1.1.1:1][00000000000000000000][10000]/23, version 2
  Paths: (1 available, best #1, table evi_1)
  Local
    :: (via default) from 0.0.0.0 (101.1.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:1001:1
      L2 srv6 in-sid: 2001:1:1:E000:: (pl:64 bl:32 nl:16 fl:16 al:0 bh:65)
BGP routing table entry for [1][101.1.1.1:1][00000000000000000000][20000]/23, version 6
  Paths: (1 available, best #1, table evi_1)
  Local, imported path from [1][101.6.6.6:1][00000000000000000000][20000]/23 (global)
    2001:101:6:6::6 (metric 40) (via default) from 2001:101:4:4::4 (101.4.4.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1001:1
      Originator: 101.6.6.6, Cluster list: 101.4.4.4
      L2 srv6 out-sid: 2001:1:6:E000:: (pl:64 bl:32 nl:16 fl:16 al:0 bh:65)
```

Key fields to verify:

L2 srv6 in-sid — The local SRv6 SID for the VPWS VC (on locally sourced routes).

L2 srv6 out-sid — The remote SRv6 SID for the VPWS VC (on imported routes).

Next Hop — Must be a valid IPv6 address of the remote PE.

Extended Community: RT — Route targets must match between PEs.

Verifying EVPN VPWS VC Status

Use the show l2vpn evpn vpws vc all command for a brief summary.

```
Device# show l2vpn evpn vpws vc all
EVPN ID Source   Target   Type   Name/Interface                  Status
------- -------- -------- ------ -------------------------------- ----------
1       10000    20000    p2p    vc_1                            up
                                 Te0/0/2:1                       up
1       10001    20001    p2p    vc_2                            up
                                 Te0/0/2:10                      up
```

Use the show l2vpn evpn vpws vc all detail command for detailed VC information including SRv6 SIDs and traffic statistics.

```
Device# show l2vpn evpn vpws vc all detail
EVPN name: vc_1, state: up, type: point-to-point
  EVPN ID: 1
  VPWS Service Instance ID: Source 10000, Target 20000
  Encapsulation: srv6
  SRv6 Locator: locator1
  SRv6 SIDs: Local 2001:1:1:E000::, Remote 2001:1:6:E000::
  Next Hop Address: 2001:101:6:6::6
  Associated member interface Te0/0/2 up, Te0/0/2:1 status is up
  Rx Counters
    266856 input transit packets, 44089591 bytes
    0 drops
  Tx Counters
    260690 output transit packets, 64988756 bytes
    0 drops
  5 VC FSM state transitions, Last 5 shown
    Prov: Idle -> Prov, Wed Mar 25 15:25:23.769 (1w2d ago)
    AdmUp: Prov -> LocWait, Wed Mar 25 15:26:25.483 (1w2d ago)
    LocUp: LocWait -> RemWait, Wed Mar 25 15:26:25.484 (1w2d ago)
    RemUp: RemWait -> Act, Wed Mar 25 15:27:48.627 (1w2d ago)
    DpUp: Act -> Est, Wed Mar 25 15:27:48.633 (1w2d ago)
```

Key fields to verify:

Encapsulation: srv6 — Confirms SRv6 transport is in use.

SRv6 Locator — Shows the locator in use for this VC.

SRv6 SIDs: Local / Remote — Displays the local and remote END.DX2 SIDs.

Next Hop Address — IPv6 address of the remote PE.

VC FSM state — Should reach Est (Established) when fully operational.

Verifying EVPN VPWS Summary

```
Device# show l2vpn evpn vpws summary
L2VPN EVPN VPWS:
  EVIs (point-to-point): 1
  Total VCs: 2
    2 up, 0 down, 0 admin-down, 0 hot-standby, 0 other
  BGP: ASN 1001, address-family l2vpn evpn configured
  Router ID: 101.1.1.1
```

Verifying L2Route EVPN EAD Routes

Use the show l2route evpn ead per-evi command to view the L2 routing information base for EAD per-EVI routes.

```
Device# show l2route evpn ead per-evi
EVI   ETag       Prod   ESI                      Next Hop(s)
----- ---------- ------ ------------------------ ----------------------------------------
1     20000      L2VPN  N/A                      S:2001:1:1:E000:: Te0/0/2:1
1     20000      BGP    N/A                      S:2001:1:6:E000:: 2001:101:6:6::6
1     20001      L2VPN  N/A                      S:2001:1:1:E001:: Te0/0/2:10
1     20001      BGP    N/A                      S:2001:1:6:E001:: 2001:101:6:6::6
```

Use the show l2route evpn ead per-evi detail command for detailed information including SID parameters.

```
Device# show l2route evpn ead per-evi detail
EVPN Instance:            1
Ethernet Tag:             20000
Producer Name:            L2VPN
ESI:                      N/A
Next Hop(s):              Te0/0/2:1
Local SID:                2001:1:1:E000::
                             PL:64  BL:32  NL:16  FL:16
                             AL:0   Static:No  Behaviour:65
EVPN Instance:            1
Ethernet Tag:             20000
Producer Name:            BGP
ESI:                      N/A
Next Hop(s):              2001:101:6:6::6
                          SID: 2001:1:6:E000::
                             PL:64  BL:32  NL:16  FL:16
                             AL:0   Static:No  Behaviour:65
```

Producer: L2VPN — Locally originated routes.

Producer: BGP — Remotely learned routes.

Behavior: 65 — Corresponds to END.DX2 with Next C-SID.

Verifying L2FIB EVPN VPWS Encap

Use the show l2fib evpn vpws encap command to verify the forwarding plane programming.

```
Device# show l2fib evpn vpws encap
Type                          : EVPN_VPWS
Reference Count               : 1
SW Handle                     : 0x7F2A27A68FA0
PathList ID                   : 1
PathList Type                 : VPWS
Ethernet Segment              : 0000.0000.0000.0000.0000
Path Count                    : 1
Paths                         : [None] evi:1 etag:20000 [2001:101:6:6::6]
OCE Type                      : SRv6
Forwarding ID                 : 0x1800023
Forwarding OCE                : 0x7F2A276ECF90
Forwarding OCE Type           : 0x4E (srv6 sid list)
next hop track flags          : PRIMARY REGISTERED RESOLVED
Created                       : 1w2d ago
```

Note: - The Forwarding ID is the label in CEF table

Key fields to verify:

OCE Type: SRv6 — Confirms SRv6 output chain is programmed.

next hop track flags: PRIMARY REGISTERED RESOLVED — Confirms the SRv6 underlay path is resolved.

Verifying L2FIB EVPN VPWS Encap

Use the show ip cef label table command to verify the forwarding plane programming.

```
Device#show ip cef label-table
Label               Next Hop             Interface
0                    no route
25165859             FE80::768F:C2FF:FE47:8E31       TenGigabitEthernet0/0/1
                     FE80::768F:C2FF:FE48:78C1       TenGigabitEthernet0/0/0
25165860             FE80::768F:C2FF:FE47:8E31       TenGigabitEthernet0/0/1
                     FE80::768F:C2FF:FE48:78C1       TenGigabitEthernet0/0/0
```

Note: The label is the forwarding id of the VPWS virtual circuit

Verifying SRv6 Platform Capabilities

This command displays the platform SRv6 capabilities, including support for END.DX2 with uSID and EVPN VPWS single homing.

```
Device#show srv6 capabilities-parameters
Platform Capabilities:
 SRv6:Yes
 PFP:Yes
 MVPN:Yes
 TILFA:Yes
 EVPN VPWS Single-Homing:Yes
 Transit-Node-PTB-Relay, Headend-Node-PTB-Relay
 Endpoint behaviors:
  uN (PSP/USD)
  uA (PSP/USD)
  uDT6
  uDT4
  uDT46
  uDX2
  End.DTMC4
 Headend behaviors:
  H.ENCAP.RED
 Encap Parameters:
  Max-SL :16 (H-ENCAP-MSD:16 SL-MSD:16, END-POP-MSD:16 END-ENCAP-MSD:16)
  Encap :Collapsed
  Hop-limit propagate :Yes
  Traffic-class propagate :Yes
  Path-MTU supported :static
  SRH mode :
   combine :Yes
   stack :No
   insert :Yes

Parameters in-use:
 Encap Parameters:
  Source Address: 2001:101:1:1::1, Loopback0 (Default)
  Hop-Limit: 64 (Default)
  Traffic-class: 0 (Default)
  MTU: not configured (Default)
```

BGP SRv6 L3VPN On-Demand Next-Hop

BGP SRv6L3VPN On-Demand Next-Hop

When redistributing routing information across domains, provisioning of multi-domain services (L3VPN) has its own complexity and scalability issues. The On-Demand Next-Hop (ODN) configuration allows BGP to dynamically create SR policies as a result of learning routes with an extended color community attribute. It then installs the replied multi-domain LSP for the duration of the service into the local forwarding information base (FIB).

This section describes how SRv6 Traffic Engineering (SRv6-TE) works with the On-Demand Next-hop (ODN) mechanism.

Prerequisites for BGP SRv6 L3VPN ODN

Refer to the SRv6 under IS-IS section before configuring BGP SRv6L3VPN On-Demand Next-Hop.

Information About BGP SRv6L3VPN ODN

The Segment Routing-Traffic Engineering (SR-TE) On-Demand Next-Hop (ODN) is a mechanism that allows the steering of traffic on a segment routing policy based on the attributes of the packets. Packets are classified using Cisco's enhanced Policy-based Routing (ePBR) and then marked with internal tags known as Forward Classes (FCs). A PFP or PDP is used to route the marked packets based on the mappings between an FC and its corresponding path. This means that the traffic is steered based on its ePBR markings and switched to the appropriate path based on the FC of the packet.

BGP Color Extended Community and VRF Prefix Coloring

In the SR-TE mechanism, the prefix that needs an SR-TE routing path is associated with a color-extended community (an attribute that assigns color to the prefixes). Currently, BGP has the capability to attach the color-extended community based only on the neighbor command routemap outbound configuration. To color the prefixes based on attributes such as Source-VRF, Destination-VRF, CE-neighbor, and Source protocol, the following ways of attaching color are introduced:

- VRF Export Coloring
- VRF Import Coloring
- Route Redistribution Coloring into BGP
- Neighbor In-bound Coloring

Route-map Additive Color Extended Community

New color-extended communities from route-map can be added to the existing color extended communities list present in the prefix attribute if the new color is not already in the list. To add the new color extended community to the existing list of color extended communities of the prefix instead of replacing the existing, the keyword **additive** is used with the **route-map set extcommunity color** command:

```
route-map SRTE-color-map permit
         set extcommunity color < 1-4294967295> [additive]
```

Note

- If the **route-map set extcommunity color** command is configured without the **additive** keyword, the new color-extended community from route-map will replace any exiting color communities present in the prefix attribute.
- The number of color-extended communities that can be added to a prefix is limited by the number of extended communities that can be added to a BGP update message.

BGP Receiving of Multiple Colors

If BGP receives an update with multiple color extended communities, it creates an SR policy only for the highest color value, and not for other color values, as defined by the draft-ietf-spring-segment-routing-policy-08.

If the SR policy of highest color is down or unavailable, the BGP path will still be the best path, but will use the optimal path as routing path (for example, SRv6 SID as routing path).

The **show ip bgp <prefix>** command includes the SR policy information for the highest color, as SR policy is only created for the highest color value. The state of the highest color SR policy decides whether the SR policy or the optimal path is used as the routing path.

Note Multiple color handling, as defined in latest drafts (draft-ietf-spring-segment-routing-policy-09, RFC 9256) to use SR Policy of a lower color value when the SR policy of higher color value is down or unavailable, is not supported.

Configuring SRv6 L3VPN ODN

The following examples show how to configure SRv6 L3VPN ODN:

PE1 (Egress) Configuration

```
route-map test permit 10
 set extcommunity color 10
!

segment-routing srv6
 locators
  locator foo
   prefix FCCC:CCC2:C3::/48
   format usid-f3216
 !
!

router bgp 1
  neighbor 2023:1::3 remote-as 1
 !
address-family vpnv4
  segment-routing srv6
   locator foo
   alloc-mode per-vrf
  exit-srv6
          !
  neighbor 2023:1::3 activate
  neighbor 2023:1::3 send-community both
```

```
  neighbor 2023:1::3 route-map test out
exit-address-family
!
```

PE2 (Ingress) Configuration

```
segment-routing traffic-eng
 on-demand color 10
  authorize
  candidate-paths
   preference 1
    constraints
     segments
      dataplane srv6
     !
    !
    dynamic

router bgp 1
  neighbor 2023:1::3 remote-as 1
 !
 address-family vpnv4
  neighbor 2023:1::3 activate
  neighbor 2023:1::3 send-community both
 exit-address-family

 address-family vpnv6
  neighbor 2023:1::3 activate
  neighbor 2023:1::3 send-community both
 exit-address-family
```

Configuring SRv6 ODN Color Template

For BGP to dynamically instantiate SR-TE SRv6 policies to steer traffic onto, on-demand next-hop (ODN) color templates are used to define the attributes of the policies. These templates already exist and are extended for SRv6 as shown in the following sample configuration:

```
ipv6 prefix-list From-PE6 seq 5 permit A006::1/128
!
segment-routing traffic-eng
 on-demand color 1000
  authorize restrict
   ipv6 prefix-list From-PE6
  candidate-paths
   preference 100
     per-flow
      forward-class 0 color 10000
      forward-class 1 color 10001
      forward-class 2 color 10002
      forward-class 3 color 10003
      forward-class 4 color 10004
  !
 on-demand color 10000
  authorize restrict
   ipv6 prefix-list From-PE6
  candidate-paths
   preference 100
    constraints
     segments
      dataplane srv6
     !
     affinity
      include-all
```

```
      name non-voice
     !
    !
   !
   dynamic
    metric
     type delay
    !
   !
  preference 50
   constraints
    segments
     dataplane srv6
    !
    affinity
     include-all
      name voice
     !
    !
   !
   dynamic
   !
  !
 !
 performance-measurement
  delay-measurement
   liveness-detection
    invalidation-action down
!
on-demand color 10001
 authorize restrict
  ipv6 prefix-list From-PE6
 candidate-paths
  preference 100
   constraints
    segments
     dataplane srv6
    !
    affinity
     include-all
      name non-voice
     !
    !
   !
   dynamic
    metric
     type delay
    !
   !
  preference 50
   constraints
    segments
     dataplane srv6
    !
    affinity
     include-all
      name voice
     !
    !
   !
   dynamic
   !
  !
 !
```

```
  performance-measurement
   delay-measurement
    liveness-detection
     invalidation-action down
```

BGP Neighbor Outbound Prefix Coloring

Use this configuration to attach a color-extended community to an outbound BGP prefix update:

```
route-map SRTE-color-map permit
      set extcommunity color <1-4294967295> [additive]

   router bgp <ASnum>
     address-family <AF>
          neighbor <address> route-map SRTE-color-map out
     exit-address-family
      !
     address-family <AF> vrf <vrfname>
          neighbor <address> route-map SRTE-color-map out
     exit-address-family
```

BGP Neighbor Inbound Prefix Coloring

Use this configuration to attach a color-extended community to an inbound BGP prefix update:

```
route-map SRTE-color-map permit
      set extcommunity color <1-4294967295> [additive]

    router bgp <ASnum>
      address-family <AF>
          neighbor <address> route-map SRTE-color-map in
      exit-address-family
      !
     address-family <AF> vrf <vrfname>
          neighbor <address> route-map SRTE-color-map in
     exit-address-family
```

BGP VRF Export Prefix Coloring

Use this configuration to attach a color extended community to the VPN prefix per the export route-map color-extended community associated with the VRF:

```
route-map SRTE-color-map permit
      set extcommunity color <1-4294967295> [additive]

   vrf def SRTE-VRF
       rd 1:1
      !
       address-family ipv4
           export map SRTE-color-map
       exit-address-family
       !
       address-family ipv6
           export map SRTE-color-map
       exit-address-family
```

BGP VRF Import Prefix Coloring

Use this configuration to attach a color extended community to the VPN prefix per the import route-map color-extended community associated with the VRF:

```
route-map SRTE-color-map permit
      set extcommunity color <1-4294967295> [additive]
```

```
   vrf def SRTE-VRF
       rd 1:1
      !
       address-family ipv4
          import map SRTE-color-map
       exit-address-family
      !
       address-family ipv6
          import map SRTE-color-map
       exit-address-family
```

BGP Route Redistribution Prefix Coloring

Use this configuration to attach a color-extended community to a BGP prefix redistributed to BGP:

```
route-map SRTE-color-map permit
      set extcommunity color <1-4294967295> [additive]

    router bgp <ASnum>
      address-family ipv4 vrf <vrf-name>
          redistribute <source-protocol> route-map SRTE-color-map
     or
          network <address> mask <network-mask> route-map SRTE-color-map
      exit-address-family
      !
      address-family ipv6 vrf <vrf-name>
          redistribute <source-protocol=> route-map SRTE-color-map
     or
          network <address>/masklen route-map SRTE-color-map
      exit-address-family
```

Verifying SRv6 L3VPN ODN Configuration

Sample Output for BGP

The **show prefix** commands display the color and binding SID associated with the BGP prefix path:

- `show bgp vpnv4 unicast vrf <vrfname> <prefix>`
- `show bgp vpnv6 unicast vrf <vrfname> <prefix>`

```
device#show bgp vpnv4 unicast vrf red 22.22.22.22
BGP routing table entry for 1:1:22.22.22.22/32, version 44
Paths: (1 available, best #1, table red)
  Advertised to update-groups:
     1
  Refresh Epoch 2
  3, imported path from 2:2:22.22.22.22/32 (global)
    2023:1::1 (via default) from 1.1.1.3 (1.1.1.3)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1 RT:2:2 Color:10
      Originator: 11.1.1.1, Cluster list: 1.1.1.3
      binding SID: 16777218 (color - 10) (state - UP)
      srv6 out-sid: FCCC:CCC1:AA88:E000::
      rx pathid: 0, tx pathid: 0x0
      Updated on Mar 29 2023 12:38:45 PST

device#show bgp vpnv6 unicast vrf red 2222::1/128
BGP routing table entry for [1:1]2222::1/128, version 45
Paths: (1 available, best #1, table red)
  Advertised to update-groups:
```

```
    1
  Refresh Epoch 2
  3, imported path from [2:2]2222::1/128 (global)
    2023:1::1 (via default) from 1.1.1.3 (1.1.1.3)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:2:2 Color:11
      Originator: 11.1.1.1, Cluster list: 1.1.1.3
      binding SID: 16777220 (color - 11) (state - UP)
      srv6 out-sid: FCCC:CCC1:AA88:E001::
      rx pathid: 0, tx pathid: 0x0
      Updated on Mar 30 2023 15:58:52 PST
```

Sample Output for SRv6 ODN Policy

The **show segment-routing traffic-eng policy name <name>** command displays the SRv6 ODN policy information:

```
device#show segment-routing traffic-eng policy name *A006::1|1000

Name: *A006::1|1000 (Color: 1000 End-point: A006::1)
  Owners : BGP
  Status:
    Admin: up, Operational: up for 00:01:16 (since 11-23 09:57:11.624)
  Candidate-paths:
    Preference 100 (BGP):
      Per-flow Information (active):
        Forward       PDP     PDP     BSID RW
          Class      Color   Status   Status
        -------- ---------- -------   --------
               0      10000      up   Pending
               1      10001      up   Pending
               2      10002      up   Pending
               3      10003      up   Pending
               4      10004      up   Pending
      Default Forward Class: 0
  Attributes:
  IPv6 caps enabled

device#show segment-routing traffic-eng policy name *A006::1|10000

Name: *A006::1|10000 (Color: 10000 End-point: A006::1)
  Owners : BGP-PFP-AUTO
  Status:
    Admin: up, Operational: up for 00:02:23 (since 11-23 09:56:34.669)
  Candidate-paths:
    Preference 100 (BGP-PFP-AUTO):
      PM State: Up
      Constraints:
        Affinity:
          include-all:
           non-voice
      Dynamic (active)
        Metric Type: DELAY, Path Accumulated Metric: 120
          F:1:1:E002:: [Adjacency-SID]
          F:1:2:E002:: [Adjacency-SID]
          F:1:3:E002:: [Adjacency-SID]
    Preference 50 (BGP-PFP-AUTO):
      PM State: Up
      Constraints:
        Affinity:
          include-all:
           voice
      Dynamic (inactive)
        Metric Type: TE, Path Accumulated Metric: 30
```

```
          F:1:4:: [Node-SID]
          F:1:5:: [Node-SID]
          F:1:6:: [Node-SID]
  Attributes:
```

The **show segment-routing traffic-eng policy name <name> detail** command displays the SRv6 ODN policy details:

```
device#show segment-routing traffic-eng policy name *A006::1|1000 detail

Name: *A006::1|1000 (Color: 1000 End-point: A006::1)
 Owners : BGP
 Status:
   Admin: up, Operational: up for 00:01:20 (since 11-23 09:57:11.624)
 Candidate-paths:
   Preference 100 (BGP):
     Per-flow Information (active):
       Forward        PDP     PDP   BSID RW
         Class      Color   Status   Status
      -------- ---------- -------  --------
             0      10000      up   Pending
             1      10001      up   Pending
             2      10002      up   Pending
             3      10003      up   Pending
             4      10004      up   Pending
    Default Forward Class: 0
  Attributes:
  IPv6 caps enabled
  Forwarding-ID: 16778667
  Per owner configs:
    BGP
      Binding SID: not configured
  Stats:
    Packets: 0  Bytes: 0

  Event history:
   Timestamp             Client          Event type          Context: Value
   ---------             ------          ----------          -------: -----
   11-23 09:56:34.668    BGP             Policy created      Name: BGP
   11-23 09:56:34.668    BGP             Set colour          Colour: 1000
   11-23 09:56:34.668    BGP             Set end point       End-point: A006::1
   11-23 09:56:34.669    BGP             Set dynamic pce     Path option: per flow
   11-23 09:57:11.624    FH Resolution   Policy state UP     Status: PFP RESOLVED CP: 100
   11-23 09:57:11.706    FH Resolution   REOPT triggered     Status: REOPTIMIZED CP: 100
   11-23 09:57:11.711    FH Resolution   REOPT triggered     Status: REOPTIMIZED CP: 100
   11-23 09:57:12.208    FH Resolution   REOPT triggered     Status: REOPTIMIZED CP: 100
```

```
device#show segment-routing traffic-eng policy name *A006::1|10000 detail

Name: *A006::1|10000 (Color: 10000 End-point: A006::1)
  Owners : BGP-PFP-AUTO
  Status:
    Admin: up, Operational: up for 00:02:28 (since 11-23 09:56:34.669)
  Candidate-paths:
    Preference 100 (BGP-PFP-AUTO):
      PM State: Up
      Constraints:
        Affinity:
          include-all:
           non-voice
      Dynamic (active)
        Metric Type: DELAY, Path Accumulated Metric: 120
          F:1:1:E002:: [Adjacency-SID]
          F:1:2:E002:: [Adjacency-SID]
          F:1:3:E002:: [Adjacency-SID]
```

```
  Preference 50 (BGP-PFP-AUTO):
    PM State: Up
    Constraints:
      Affinity:
        include-all:
         voice
    Dynamic (inactive)
      Metric Type: TE, Path Accumulated Metric: 30
        F:1:4:: [Node-SID]
        F:1:5:: [Node-SID]
        F:1:6:: [Node-SID]
Attributes:
Forwarding-ID: 16778668
Per owner configs:
  BGP-PFP-AUTO
    Binding SID: not configured
    Performance-measurement:
      liveness-detection
      invalidation-action down
Stats:
  Packets: 0  Bytes: 0
PM profile: Not configured

Event history:
  Timestamp            Client          Event type           Context: Value
  ---------            ------          ----------           -------: -----
  11-23 09:56:34.669   BGP-PFP-AUTO    Policy created       Name: BGP-PFP-AUTO
  11-23 09:56:34.669   BGP-PFP-AUTO    Set colour           Colour: 10000
  11-23 09:56:34.670   BGP-PFP-AUTO    Set end point        End-point: A006::1
  11-23 09:56:34.670   BGP-PFP-AUTO    Set dynamic          Path option: dynamic
  11-23 09:56:34.672   BGP-PFP-AUTO    Set dynamic          Path option: dynamic
  11-23 09:56:34.673   BGP-PFP-AUTO    Set delay measure    status: Enabled
  11-23 09:56:34.673   BGP-PFP-AUTO    Set Live Detection   status: Enabled
  11-23 09:56:34.673   BGP-PFP-AUTO    Set Live Invalidatio action: down
  11-23 09:57:07.662   FH Resolution   Liveness             CP: 100, SL2153 is Waiting
  11-23 09:57:07.663   FH Resolution   Liveness             CP: 50, SL2154 is Waiting
  11-23 09:57:11.609   PM              Liveness             CP: 50, SL2154 is Up
  11-23 09:57:11.623   FH Resolution   Policy state UP      Status: PATH RESOLVED CP: 50
  11-23 09:57:12.740   PM              Liveness             CP: 100, SL2153 is Up
  11-23 09:57:12.766   FH Resolution   REOPT triggered      Status: REOPTIMIZED CP: 100
  11-23 09:57:18.334   FH Resolution   REOPT triggered      Status: REOPTIMIZED CP: 100
  11-23 09:57:18.382   PM              Liveness             CP: 50, SL2154 is Unknown
  11-23 09:57:19.922   FH Resolution   REOPT triggered      Status: REOPTIMIZED CP: 100
  11-23 09:57:28.707   FH Resolution   REOPT triggered      Status: REOPTIMIZED CP: 100
  11-23 09:57:28.708   FH Resolution   Liveness             CP: 50, SL2154 is Waiting
  11-23 09:57:28.709   FH Resolution   REOPT triggered      Status: REOPTIMIZED CP: 100
  11-23 09:57:32.000   PM              Liveness             CP: 50, SL2154 is Up
  11-23 09:57:32.003   FH Resolution   REOPT triggered      Status: REOPTIMIZED CP: 100
```

For more information about verifying SRv6-TE configuration, see section *Verifying SRv6-TE Configuration*.

Debugging SRv6 L3VPN ODN Configuration

Use the following debug command to track the events related to SR ODN:

- **debug ip bgp sr-policy**

```
*Apr 10 17:35:48.773: BGP(4): 2023:1::3 rcvd UPDATE w/ attr: nexthop 2023:1::1, origin ?,
localpref 100, metric 0, originator 11.1.1.1, clusterlist 1.1.1.3, merged path 3, AS_PATH
, extended community RT:1:1 RT:2:2 Color:10, PrefixSid attribute: SRV6 SID FCCC:CCC1:AA88::
*Apr 10 17:35:48.773: BGP(4): 2023:1::3 rcvd 2:2:22.22.22.22/32, label 2162163712 (0x80E00000)
*Apr 10 17:35:48.773: BGP SRv6 SID ATTR: blk 32 node 16 fun 16 arg 0 pos 16 off 48
*Apr 10 17:35:48.774: BGP-SR Policy (7F7911708510): Binding SID 10/2023:1::1/ request
```

```
*Apr 10 17:35:48.774: BGP(4): Revise route installing 1 of 1 routes for 22.22.22.22/32 ->
0.0.0.0(red) to red IP table

PE2#show bgp vpnv4 uni vrf red 22.22.22.22
BGP routing table entry for 1:1:22.22.22.22/32, version 33
Paths: (1 available, best #1, table red)
  Advertised to update-groups:
     3
  Refresh Epoch 1
  3, imported path from 2:2:22.22.22.22/32 (global)
    2023:1::1 (via default) from 1.1.1.3 (1.1.1.3)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1 RT:2:2 Color:10
      Originator: 11.1.1.1, Cluster list: 1.1.1.3
      binding SID: 16777219 (color - 10) (state - UP)
      srv6 out-sid: FCCC:CCC1:AA88:E000::
      rx pathid: 0, tx pathid: 0x0
      Updated on Apr 10 2023 09:35:48 PST

PE2(config)#segment-routing traffic-eng
PE2(config-srte-on-demand-color)#on-demand color 10

PE2(config-srte-on-demand-color)#no authorize
*Apr 10 17:37:25.964: BGP SR: color change notification callback for color 10, auth_type 3
*Apr 10 17:37:25.964: BGP SR: color change state handler for color 10, color_state 2
*Apr 10 17:37:25.964: BGP SR: deletion of policy *2023:1::1|10 is successful
*Apr 10 17:37:25.964: BGP SR: sr_policymgr color 10, delete: timer started
*Apr 10 17:37:26.678: BGP SR: Policy change timer expired.: bgp_sr_policy_service_change
started
*Apr 10 17:37:26.679: BGP(4): Revise route installing 1 of 1 routes for 5.5.6.8/32 ->
0.0.0.0(red) to red IP table
*Apr 10 17:37:26.680: BGP(4): Revise route installing 1 of 1 routes for 10.1.1.0/24 ->
0.0.0.0(red) to red IP table
*Apr 10 17:37:26.680: BGP(4): Revise route installing 1 of 1 routes for 22.22.22.22/32 ->
0.0.0.0(red) to red IP table
PE2(config-srte-on-demand-color)#do sh bgp vpnv4 uni vrf red 22.22.22.22
BGP routing table entry for 1:1:22.22.22.22/32, version 30
Paths: (1 available, best #1, table red)
  Advertised to update-groups:
     3
  Refresh Epoch 1
  3, imported path from 2:2:22.22.22.22/32 (global)
    2023:1::1 (via default) from 1.1.1.3 (1.1.1.3)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1 RT:2:2 Color:10
      Originator: 11.1.1.1, Cluster list: 1.1.1.3
      binding SID: none (color-10)
      srv6 out-sid: FCCC:CCC1:AA88:E000::
      rx pathid: 0, tx pathid: 0x0
      Updated on Apr 10 2023 09:35:48 PST


PE2(config-srte-on-demand-color)#authorize
*Apr 10 17:37:41.096: BGP SR: color change notification callback for color 10, auth_type 2
*Apr 10 17:37:41.096: BGP SR: color change state handler for color 10, color_state 1
*Apr 10 17:37:41.096: BGP SR POLICY: policy *2023:1::1|10 create request
*Apr 10 17:37:41.097: BGP SR: sr_policymgr color 10, add
*Apr 10 17:37:41.097: BGP SR: Policy change for *2023:1::1|10, type 5, bsid 0
*Apr 10 17:37:41.097: BGP SR Policy Change notification: color: 10, nexthop: 2023:1::1:
timer started
*Apr 10 17:37:41.097: BGP_SR_POLICY: policy *2023:1::1|10 registered for policy notification
*Apr 10 17:37:41.097: BGP SR: Policy change for *2023:1::1|10, type 3, bsid 1000003
*Apr 10 17:37:41.097: BGP SR Policy Change notification: color: 10, nexthop: 2023:1::1
*Apr 10 17:37:41.097: BGP SR Policy Found: color: 10, nexthop: 2023:1::1: timer already
```

```
running
*Apr 10 17:37:41.097: BGP SR: Policy change for *2023:1::1|10, type 4, bsid 1000003
*Apr 10 17:37:41.097: BGP SR Policy Change notification: color: 10, nexthop: 2023:1::1
*Apr 10 17:37:41.097: BGP SR Policy Found: color: 10, nexthop: 2023:1::1: timer already
running
*Apr 10 17:37:42.039: BGP SR: Policy change timer expired.: bgp_sr_policy_service_change
started
*Apr 10 17:37:42.040: BGP(4): Revise route installing 1 of 1 routes for 5.5.6.8/32 ->
0.0.0.0(red) to red IP table
*Apr 10 17:37:42.040: BGP(4): Revise route installing 1 of 1 routes for 10.1.1.0/24 ->
0.0.0.0(red) to red IP table
*Apr 10 17:37:42.040: BGP(4): Revise route installing 1 of 1 routes for 22.22.22.22/32 ->
0.0.0.0(red) to red IP table
PE2(config-srte-on-demand-color)#do sh bgp vpnv4 uni vrf red 22.22.22.22
BGP routing table entry for 1:1:22.22.22.22/32, version 33
Paths: (1 available, best #1, table red)
  Advertised to update-groups:
     3
  Refresh Epoch 1
  3, imported path from 2:2:22.22.22.22/32 (global)
    2023:1::1 (via default) from 1.1.1.3 (1.1.1.3)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1 RT:2:2 Color:10
      Originator: 11.1.1.1, Cluster list: 1.1.1.3
      binding SID: 16777219 (color - 10) (state - UP)
      srv6 out-sid: FCCC:CCC1:AA88:E000::
      rx pathid: 0, tx pathid: 0x0
  Updated on Apr 10 2023 09:35:48 PST
```

For information about troubleshooting and debugging SRv6-TE, see the Troubleshooting and Debugging SRv6-TE section.

SRv6 Traffic Engineering Policies

SRv6 Traffic Engineering Policies

From Cisco IOS XE 17.12.1a, the Segment Routing Traffic Engineering (SR-TE) mechanism is extended to Segment Routing over IPv6 (SRv6).

Restrictions for SRv6-TE Policies

- From Cisco IOS XE 17.16.1a, local path computation and PCE delegated path computation are supported; PCE-initiated polices are not supported.
- Only dynamic segment-lists are supported; explicit segment-lists are not supported.
- SRv6 Binding SIDs are not supported.
- On-demand next-hop (ODN) is not supported.
- L2VPN over SR-TE is not supported.
- Auto-route announce over PFP or PDP is not supported.
- When you create a policy with multiple SIDs, the final SID to reach the egress PE is the node SID, and this will be removed from the SID list.
- The VPN SID must always have the locator information as part of the SIDs.

Information About SRv6-TE Policies

The SRv6 Traffic Engineering (SRv6-TE) uses a SRv6 policy to steer traffic through the network. The SRv6 policy includes Per-flow policies (PFP) and Per-destination policies (PDP), both of which are supported.

An ePBR policy is applied to the ingress interface to define how traffic is classified and associated with the forward-class (FC), PFP is configured with a Per-flow forward-class table up to eight entries. Each entry is indexed by a FC and points to a PDP.

For PFP, the packets are classified on the ingress interface and choose different PDP paths to forward to the same destination based on the classification by ePBR.

Configuring SRv6-TE

The following examples demonstrate how to configure locally computed SRv6-TE policies..

Configuring PDP

```
segment-routing traffic-eng
    policy SRV6PM
      color 1 end-point C02:1::1
      candidate-paths
       preference 1
        constraints
         segments
          dataplane srv6
         !
        !
        dynamic
        !
       !
      !
       preference 2
        constraints
         segments
          dataplane srv6
         !
         affinity
          exclude-any
           name blue
          !
         !
        !
        dynamic
           metric
              type delay
         !
        !
      performance-measurement
       delay-measurement
        liveness-detection
         invalidation-action down
        !
       !
      !
```

Configuring PFP

```
segment-routing traffic-eng
    policy PFP
      color 100 end-point C02:1::1
```

```
      candidate-paths
       preference 1
        per-flow
         forward-class 0 color 1
         forward-class 1 color 2
         forward-class 2 color 3
         forward-class 3 color 4
         forward-class 4 color 5
```

Configuring ePBR

```
policy-map type epbr PFP
 class FC1
  set forward-class 1
 class FC2
  set forward-class 2
 class FC3
  set forward-class 3
 class FC4
  set forward-class 4
 class class-default
  set forward-class 0

interface TenGigabitEthernet2/2/0.1000
 encapsulation dot1Q 1000
  vrf forwarding vpn-1000
   ip address 17.0.0.1 255.255.255.0
   ipv6 address 1700::1/64
   service-policy type epbr input PFP
```

Configuring Static Route

1. IPv6 static route for a prefix, NO SR policy, and optional VPN SID

   ```
   ipv6 route vrf blue 1002:1::/64 2001:1::2 nexthop-vrf default sid-list h-encaps-red
   FCCC:CCC1:C3:E005::
   ```

2. IPv6 static route for a prefix with traffic steered via optional SR policy and VPN SID

   ```
   ipv6 route vrf blue 1002:1::/64 segment-routing srv6 via policy PFP sid-list h-encaps-red
    FCCC:CCC1:C3:E005::
   ```

3. IPv4 static route for a prefix with traffic steered via optional SR policy and VPN SID

   ```
   ip route vrf blue 2.2.2.2 255.255.255.255 segment-routing srv6 via policy PFP sid-list
   h-encaps-red FCCC:CCC1:C3:E004::
   ```



Note IPv4 static route for a prefix, NO SR policy, and optional VPN SID is NOT supported.

Configuring PCE Delegation for Path Computation

The following sample configuration demonstrates how to change the locally computed SRv6-TE PDP in the previous example to be delegated for path computation to a PCE. From Cisco IOS XE 17.16.1a, you can configure PCE delegation for path computation.

```
segment-routing traffic-eng
    pcc
      pce address C5:1::1
    policy SRV6PM
```

```
     color 1 end-point C02:1::1
      candidate-paths
        preference 1
         constraints
           segments
            dataplane srv6
!
!
    dynamic
      pcep
!
!
!
        preference 2
         constraints
            segments
             dataplane srv6
!
    affinity
     exclude-any
      name blue
!
!
!
    dynamic
       pcep
        metric
         type delay
!
!
    performance-measurement
     delay-measurement
      liveness-detection
       invalidation-action down
!
!
!
```

Verifying SRv6-TE Configuration

Example 1: Use the **show segment-routing traffic-eng policy name** command to verify SRv6-TE configuration, with PDP and PFP:

```
router# show segment-routing traffic-eng policy name SRV6PM detail

Name: SRV6PM (Color: 1 End-point: C02:1::1)
  Owners : CLI
  Status:
    Admin: up, Operational: up for 70:55:04 (since 04-11 12:10:05.054)
  Candidate-paths:
    Preference 2 (CLI):
      PM State: Up
      Constraints:
        Affinity:
          exclude-any:
           blue
      Dynamic (active)
        Metric Type: DELAY, Path Accumulated Metric: 40
          FCCC:CCC1:AA22:: [Node-SID]
          FCCC:CCC1:AA33:: [Node-SID]
          FCCC:CCC1:AA11:: [Node-SID]
          FCCC:CCC1:AA11:E001:: [Adjacency-SID]
    Preference 1 (CLI):
```

```
      PM State: Unknown
      Dynamic (inactive)
        Inactive Reason: Perf Measure State Change to Pending
        Metric Type: TE, Path Accumulated Metric: 10
          FCCC:CCC1:C3:: [Node-SID]
  Attributes:
  Forwarding-ID: 16777217
  Per owner configs:
    CLI
      Binding SID: not configured
      Performance-measurement:
        liveness-detection
        invalidation-action down
  Stats:
    Packets: 0  Bytes: 0
  PM profile: Not configured
```

```
router# show segment-routing traffic-eng policy name PFP

Name: PFP (Color: 100 End-point: C02:1::1)
  Owners : CLI
  Status:
    Admin: up, Operational: up for 00:03:00 (since 04-17 10:46:06.552)
  Candidate-paths:
    Preference 1 (CLI):
      Per-flow Information (active):
        Forward        PDP     PDP  BSID RW
          Class      Color  Status   Status
       -------- ---------- ------- --------
              0          1      up  Pending
              1          2      up  Pending
      Default Forward Class: 0
  Attributes:
```

Example 2: Use the **show ip cef label-table <label> internal** command to view the PFP label details.

```
router# show ip cef label-table 16777218 internal
Label-FIB is Enabled
VRF Default
 3 prefixes (3/0 fwd/non-fwd)
 Table id 0x30000000
 Database epoch:        0 (3 entries at this epoch)

16777218 , epoch 0, refcnt 8, per-destination sharing
  sources: RR, Bnd-Lbl-SRv6-Pol
  subblocks:
    1 RR source [no flags]
    Binding Label SRv6 Policy: 16777218
      Policy-Name: PFP (16777218)  è PFP Policy Name
      Path: 0
        Flags: 00000000
        IPv6 Header Parameters
          TC: 0   Flow: 0       Hop Limit: 0
          Src: C01:1::1
          Dst: 16777217
        Segment List (0)
      Path: 1
        Flags: 00000000
        IPv6 Header Parameters
          TC: 0   Flow: 0       Hop Limit: 0
          Src: C01:1::1
          Dst: 16777217
```

```
    Segment List (0)
  Path: 2
    Flags: 00000000
    IPv6 Header Parameters
      TC: 0   Flow: 0       Hop Limit: 0
      Src: C01:1::1
      Dst: 16777217
    Segment List (0)
  Path: 3
    Flags: 00000000
    IPv6 Header Parameters
      TC: 0   Flow: 0       Hop Limit: 0
      Src: C01:1::1
      Dst: 16777217
    Segment List (0)
  Path: 4
    Flags: 00000000
    IPv6 Header Parameters
      TC: 0   Flow: 0       Hop Limit: 0
      Src: C01:1::1
      Dst: 16777217
    Segment List (0)
  Path: 5
    Flags: 00000000
    IPv6 Header Parameters
      TC: 0   Flow: 0       Hop Limit: 0
      Src: C01:1::1
      Dst: 16777217
    Segment List (0)
  Path: 6
    Flags: 00000000
    IPv6 Header Parameters
      TC: 0   Flow: 0       Hop Limit: 0
      Src: C01:1::1
      Dst: 16777217
    Segment List (0)
  Path: 7
    Flags: 00000000
    IPv6 Header Parameters
      TC: 0   Flow: 0       Hop Limit: 0
      Src: C01:1::1
      Dst: 16777217
    Segment List (0)
```

Example 3: Use the **show segment-routing traffic-eng cspf** command to view the CSPF details.

```
router# show segment-routing traffic-eng cspf ipv6 source A001::1 destination A006::1
metric-type delay

Path:
  HOP0: SRv6 NODE SID=F:1:6::
Path Cost = 10

CSPF result: Shortest Path Success (rc=8)
```

Example 4: Use the show prefix commands to display the color and binding SID associated with the BGP prefix path:

```
router# show bgp vpnv4 unicast vrf red 22.22.22.22
BGP routing table entry for 1:1:22.22.22.22/32, version 14
Paths: (1 available, best #1, table red)
Advertised to update-groups:
```

```
3
Refresh Epoch 1
3, imported path from 2:2:22.22.22.22/32 (global)
2023:1::1 (via default) from 1.1.1.3 (1.1.1.3)
Origin incomplete, metric 0, localpref 100, valid, internal, best
Extended Community: RT:1:1 RT:2:2 Color:10
Originator: 11.1.1.1, Cluster list: 1.1.1.3
binding SID: 16777217 (color - 10) (state - UP)
srv6 out-sid: FCCC:CCC1:AA88:E000::
rx pathid: 0, tx pathid: 0x0
Updated on Jun 12 2023 15:33:20 PST
```

```
router# show bgp vpnv6 unicast vrf red 2222::1/128
BGP routing table entry for [1:1]2222::1/128, version 13
Paths: (1 available, best #1, table red)
Advertised to update-groups:
3
Refresh Epoch 1
3, imported path from [2:2]2222::1/128 (global)
2023:1::1 (via default) from 1.1.1.3 (1.1.1.3)
Origin incomplete, metric 0, localpref 100, valid, internal, best
Extended Community: RT:2:2 Color:10
Originator: 11.1.1.1, Cluster list: 1.1.1.3
binding SID: 16777217 (color - 10) (state - UP)
srv6 out-sid: FCCC:CCC1:AA88:E001::
rx pathid: 0, tx pathid: 0x0
Updated on Jun 12 2023 15:33:20 PST
```

Example 5: Use the **show pce client peer** command to verify the PCEP connections:

```
router# show pce client peer
       PCC's peer database:
       --------------------
       Peer address: 192.168.0.6, Precedence: 255
       Client POLICY_MGR
        State up
        Capabilities: Stateful, Update, Segment-Routing, Instantiation, SRv6
```

Troubleshooting and Debugging SRv6-TE

Use the following commands to troubleshoot SRv6TE:

- **debug ip bgp sr-policy**
- **debug segment-routing traffic-eng**
  - **forwarding**: SR forwarding debug
  - **ha**: SR High-Availability debug
  - **path**: SR path debug
  - **pcalc**: SR pcalc debug
  - **policy**: SR policy debug
  - **topology**: SR topology debug

```
router# debug ip bgp sr-policy
*Apr 10 17:35:48.773: BGP(4): 2023:1::3 rcvd UPDATE w/ attr: nexthop 2023:1::1, origin ?,
localpref 100, metric 0, originator 11.1.1.1, clusterlist 1.1.1.3, merged path 3, AS_PATH
```

```
, extended community RT:1:1 RT:2:2 Color:10, PrefixSid attribute: SRV6 SID FCCC:CCC1:AA88::
*Apr 10 17:35:48.773: BGP(4): 2023:1::3 rcvd 2:2:22.22.22.22/32, label 2162163712 (0x80E00000)
*Apr 10 17:35:48.773: BGP SRv6 SID ATTR: blk 32 node 16 fun 16 arg 0 pos 16 off 48
*Apr 10 17:35:48.774: BGP-SR Policy (7F7911708510): Binding SID 10/2023:1::1/ request
*Apr 10 17:35:48.774: BGP(4): Revise route installing 1 of 1 routes for 22.22.22.22/32 ->
0.0.0.0(red) to red IP table
```

Performance Measurement for SRv6

Performance Measurement for SRv6

Feature Name	Release Information	Description
Performance Measurement for Segment Routing over IPv6	Cisco IOS XE Dublin 17.12.1a	This feature extends the performance measurement liveness to Segment Routing configuration over IPv6 data plane.

Performance Measurement Liveness for SRv6

From Cisco IOS XE 17.12.1a, Performance Measurement liveliness is extended to Segment Routing over IPv6 dataplane.

This feature enables Performance Measurement (PM) liveness detection for an SR policy on all the segment lists of every candidate path that are present in the forwarding table using PM probes. You can monitor the traffic path and efficiently detect any drop of traffic due to cable or hardware or configuration failures.

Prerequisites

- SRV6 must be enabled on all nodes before configuring PM for SRv6.

Restrictions

- By default, probes are sent every 3 seconds. You can increase the interval using the burst interval parameter (*). Reducing the probe interval to below 3 seconds is not recommended.

Configuring PM Liveness for SRv6

Use the following examples to configure PM liveliness for SRv6.

- Use the **liveness-detection** configuration under an SRV6 Policy to continuously monitor the state of SRV6 paths. This option provides only monitoring; no action is taken by the Policy Manager.
- Use the **invalidation-action down** configuration to configure the Policy Manager to:
  - Have the path programmed in HW only after it was validated with PM probes.
  - Continuously monitor the path.
  - Reoptimize to a different CP if PM probes stop working,

    or

bring the policy down if no other path is available.

Configure SRv6-TE PM Liveness under Policy

```
policy SRV6PM
  performance-measurement
   delay-measurement
    liveness-detection
     invalidation-action down
```

Configure Default Delay Profile for Liveness

```
performance-measurement
 delay-profile sr-policy
  probe
   liveness-detection
    multiplier 3
```

The following sections describe the recommended configurations for scaling deployment.

Configure PM Punt Policer for all PE Nodes

```
platform punt-policer sr-twamp-probe 3000
platform punt-policer sr-twamp-probe 3000 high
performance-measurement
  max-pps 3000
```

Configure Interface Queue for WAN Interface on Headend and Endpoint Nodes

```
interface Tunnel121
  hold-queue 10000 in
```

OR

```
interface GigabitEthernet0/0/1
  hold-queue 10000 in
```



Note The WAN interface includes physical interfaces and GRE-TP tunnels.

Verifying Performance Measurement for SRv6

Use the following show commands to verify PM configuration for SRv6.

Example 1: **`show performance-measurement sr-policy name <name>`**

```
device# show performance-measurement sr-policy name SRV6PM
SR Policy name: SRV6PM
  Color                          : 1
  Endpoint                       : C02:1::1
  Source                         : C01:1::1
  Profile name                   : Not configured
  Policy Update Timestamp        : 04-11 12:12:51.658
  Number of candidate-paths      : 2
  Candidate-Path:
    Preference                   : 1
    Protocol-origin              : CLI
    Discriminator                : 0
```

```
    Number of segment-lists       : 1
    Number of atomic paths        : 1
    Number of live UP atomic paths: 0
    Number of live Unknown atomic : 0
    Max Pkts per Burst            : 1500
    Max Pkts per Probe            : 15000
    AP Min Run per Probe          : 3
    Round-robin bursts            : 1
    Round-robin probes            : 1
    Last advertisement:
      Advertised at: 12:12:06 04-11 2023 (516007 seconds ago)
Atomic path:
        Hops                      : C2:1::1, C3:1::1, C1:1::1
                                  : 2021:2::1
        Labels                    : FCCC:CCC1:AA22:AA33:AA11:E004::
        Outgoing Interface        : Ethernet0/2
        Max IP MTU                : 1500
        Next Hop                  : FE80::A8BB:CCFF:FE00:FA10
        Destination               : C02:1::1
        Session ID                : 8
        Last advertisement:
          No advertisements have occured
        Next advertisement:
          Aggregated delays (uSec): avg: 2744, min: 1480, max: 21676, variance: 1172
          Rolling average (uSec): 2744
        Last probe:
          Packets Sent: 10, received: 10
          Measured delays (uSec): avg: 1666, min: 1480, max: 1853, variance: 186
        Current probe:
          Packets Sent: 2, received: 2
          Measured delays (uSec): avg: 6192, min: 1619, max: 10765, variance: 4573
        Probe samples:
          Packet Rx Timestamp       Measured Delay (nsec)
          11:37:29 04-17 2023       1619000
          11:37:26 04-17 2023       10765000
```

Example 2: **`show performance-measurement sr-policy name <name> d p v | s Liveness`**

```
device# show performance-measurement sr-policy name SRV6PM d p v | s Liveness
        Liveness Detection:
          Session Creation Timestamp: 04-11 12:10:49.981
          Session State: Down
          Last State Change Timestamp: 04-11 12:12:51.656
          Missed count   [consecutive]: 84752
          Received count [consecutive]: 0
          Backoff                     : 1
          Unique Path Name            : Path-10
          Loss in Last Interval       : 100 % [TX: 7 RX: 0]
        Liveness Detection:
          Session Creation Timestamp: 04-11 12:12:36.636
          Session State: Up
          Last State Change Timestamp: 04-11 12:12:36.728
          Missed count   [consecutive]: 0
          Received count [consecutive]: 84717
          Backoff                     : 0
          Unique Path Name            : Path-12
          Loss in Last Interval       : 0 % [TX: 7 RX: 7]
        Liveness Detection:
          Session Creation Timestamp: 04-11 12:12:36.636
          Session State: Up
          Last State Change Timestamp: 04-11 12:12:36.728
          Missed count   [consecutive]: 0
          Received count [consecutive]: 84717
```

```
          Backoff                     : 0
          Unique Path Name            : Path-13
          Loss in Last Interval       : 0 % [TX: 7 RX: 7]
```

Example 3: `show segment-routing traffic-eng policy all type per-destination`

```
device# show segment-routing traffic-eng policy all type per-destination
Name: SRV6PM (Color: 1 End-point: C02:1::1)
  Owners : CLI
  Status:
    Admin: up, Operational: up for 70:55:04 (since 04-11 12:10:05.054)
  Candidate-paths:
    Preference 2 (CLI):
      PM State: Up
      Constraints:
        Affinity:
          exclude-any:
           blue
      Dynamic (active)
        Metric Type: DELAY, Path Accumulated Metric: 40
          FCCC:CCC1:AA22:: [Node-SID]
          FCCC:CCC1:AA33:: [Node-SID]
          FCCC:CCC1:AA11:: [Node-SID]
          FCCC:CCC1:AA11:E001:: [Adjacency-SID]
    Preference 1 (CLI):
      PM State: Unknown
      Dynamic (inactive)
        Inactive Reason: Perf Measure State Change to Pending
        Metric Type: TE, Path Accumulated Metric: 10
          FCCC:CCC1:C3:: [Node-SID]
  Attributes:
```

Example 4: `show performance-measurement history interfaces adv`

```
device# show performance-measurement history interfaces adv
Interface Name: Ethernet0/0 (ifh: 0x2)
  Delay-Measurement history (uSec):
    Session ID: 1
    Advertisement Timestamp   Average   Min       Max       Action
    12:10:05 04-11 2023       10        10        10        CFG
Interface Name: Ethernet0/1 (ifh: 0x3)
  Delay-Measurement history (uSec):
    Session ID: 2
    Advertisement Timestamp   Average   Min       Max       Action
    12:10:05 04-11 2023       15        15        15        CFG
Interface Name: Tunnel100 (ifh: 0x15)
  Delay-Measurement history (uSec):
    Session ID: 3
    Advertisement Timestamp   Average   Min       Max       Action
    13:10:55 04-13 2023       603       307       969       PER-MIN
    13:04:46 04-13 2023       8696      1384      18908     PER-MIN
    10:31:05 04-13 2023       6897      377       38335     PER-MIN
    10:26:56 04-13 2023       6792      1802      13778     PER-MIN
    12:12:26 04-11 2023       3018      363       14081     FIRST
Interface Name: Tunnel101 (ifh: 0x16)
  Delay-Measurement history (uSec):
    Session ID: 4
    Advertisement Timestamp   Average   Min       Max       Action
    12:12:16 04-11 2023       1841      263       8400      FIRST
```

Example 5: `show performance-measurement history sr-policy liveness-notification`

```
device# show performance-measurement history sr-policy liveness-notification
SR Policy name: pdp-voice
  Candidate-Path:
    Preference                : 10
    Protocol-origin           : CLI
    Discriminator             : 0
    Active                    : No
    Segment-list:
      Name                    : SL13
      Atomic path:
        Hops                  : A006::1
        Labels                : ::
        Outgoing Interface    : Tunnel16
        Next Hop              : 1634::6
        Destination           : A006::1
        Delay-Measurement:
          Session ID          : 16
          Liveness state change timestamp                   New State
          04:20:25 01-15 2023                               Up

  Candidate-Path:
    Preference                : 50
    Protocol-origin           : CLI
    Discriminator             : 0
    Active                    : No
    Segment-list:
      Name                    : SL12
      Atomic path:
        Hops                  : ::, ::, 5646::5
        Labels                : F:1:2:5:E003::
        Outgoing Interface    : Tunnel12
        Next Hop              : 1211::2
        Destination           : A006::1
        Delay-Measurement:
          Session ID          : 23
          Liveness state change timestamp                   New State
          04:30:19 01-15 2023                               Up

  Candidate-Path:
    Preference                : 100
    Protocol-origin           : CLI
    Discriminator             : 0
    Active                    : Yes
    Segment-list:
      Name                    : SL11
      Atomic path:
        Hops                  : ::, ::, 5631::5
        Labels                : F:1:4:5:E002::
        Outgoing Interface    : GigabitEthernet2
        Next Hop              : FE80::5054:FF:FE1A:DD62
        Destination           : A006::1
        Delay-Measurement:
          Session ID          : 14
          Liveness state change timestamp                   New State
          04:20:17 01-15 2023                               Up
```

Example 6: `show isis teapp`

```
device# show isis teapp
Tag null:
   ISIS TEAPP Information: Topology(ID:0x0) Type:SRTE, Enabled:1, Router ID:0.0.0.0 IPv6
Router ID:C01:1::1
 Topology Id:0x0 Teapp_type:SRTE
```

```
   Interface(hdl:0x2): Ethernet0/0
    Affinity: set 1, affinity_bits 8
    TE Metric: set 1, te_metric 1000
    Extended Affinity: set 1, length 1, ext_affinity_bits: 8
 Topology Id:0x0 Teapp_type:SRTE
   Interface(hdl:0x3): Ethernet0/1
    Affinity: set 1, affinity_bits 8
    TE Metric: set 1, te_metric 1000
    Extended Affinity: set 1, length 1, ext_affinity_bits: 8
   ISIS TE Attr PM Information:
    Et0/0: IDB num:2 Min:10 Max:10 Min-max-anomaly:0 Avg:10 Avg-anomaly:0 Var:0
          Is-Loss-set:0 Loss:0 Loss-anomaly:0
    Et0/1: IDB num:3 Min:15 Max:15 Min-max-anomaly:0 Avg:15 Avg-anomaly:0 Var:0
          Is-Loss-set:0 Loss:0 Loss-anomaly:0
    Tu100: IDB num:21 Min:307 Max:969 Min-max-anomaly:0 Avg:603 Avg-anomaly:0 Var:109
          Is-Loss-set:0 Loss:0 Loss-anomaly:0
    Tu101: IDB num:22 Min:263 Max:8400 Min-max-anomaly:0 Avg:1841 Avg-anomaly:0 Var:1042
          Is-Loss-set:0 Loss:0 Loss-anomaly:0
device#
```

Example 7: `show performance-measurement responder summary`

```
device# show performance-measurement responder summary
  Total interfaces                              : 5
  Total query packets received                  : 509200
  Total reply packets sent                      : 509200
  Total reply packets sent errors               : 0
  Total URO TLV not present errors              : 0
  Total invalid port number errors              : 0
  Total no source address errors                : 0
  Total no return path errors                   : 0
  Total unsupported querier control code errors : 0
  Total unsupported timestamp format errors     : 0
  Total timestamp not available errors          : 0
  Total unsupported mandatory TLV errors        : 0
  Total invalid packet errors                   : 0
  Total loss probe color errors                 : 0
  Current rate                                  : 1 pkts/sec
  Rate high water mark                          : 3 pkts/sec
```

Example 8: `show monitor event-trace perf_measure all`

```
device# show monitor event-trace perf_measure all
Perf Measure error events:
Perf Measure event events:
*Apr 11 17:10:05.115: PM-TRACE-IGP-ADV :flood Ethernet0/0 10 10 10
*Apr 11 17:10:05.116: PM-TRACE-IGP-ADV :flood Ethernet0/1 15 15 15
*Apr 11 17:12:16.492: PM-TRACE-IGP-ADV :flood Tunnel101 263 8400 1841
*Apr 11 17:12:26.582: PM-TRACE-IGP-ADV :flood Tunnel100 363 14081 3018
*Apr 13 15:26:56.861: PM-TRACE-IGP-ADV :flood Tunnel100 1802 13778 6792
*Apr 13 15:31:05.510: PM-TRACE-IGP-ADV :flood Tunnel100 377 38335 6897
*Apr 13 18:04:46.608: PM-TRACE-IGP-ADV :flood Tunnel100 1384 18908 8696
*Apr 13 18:10:55.245: PM-TRACE-IGP-ADV :flood Tunnel100 307 969 603
interrupt context allocation count = 0
```

Example 9: `show performance-measurement summary`

```
device# show performance-measurement summary
Total interfaces                            : 4
Total SR Policies                           : 2
Total endpoints                             : 0
Maximum PPS                                 : 2000 pkts/sec
```

```
Dual-color gre bit-position                  : 9 - Failed, last success 0
Interface Delay-Measurement:
  Total sessions                             : 4
  Counters:
    Packets:
      Total sent                             : 338865
      Total received                         : 338861
    Errors:
      Total sent errors                      : 14
      Total received errors                  : 0
    Probes:
      Total started                          : 33892
      Total completed                        : 33884
      Total incomplete                       : 4
      Total advertisements                   : 8
SR Policy Delay-Measurement:
  Total sessions                             : 4
  Counters:
    Packets:
      Total sent                             : 339076
      Total received                         : 169602
    Errors:
      Total sent errors                      : 0
      Total received errors                  : 0
    Probes:
      Total started                          : 33912
      Total completed                        : 16964
      Total incomplete                       : 16948
      Total advertisements                   : 243
```

SRv6 OAM

SRv6 Operations, Administration, and Maintenance

From Cisco IOS XE 17.12.1a, Operations, Administration, and Maintenance (OAM) functionality is suported by SRv6, using Segment Lists and SRv6 Policy.

Restrictions for SRv6

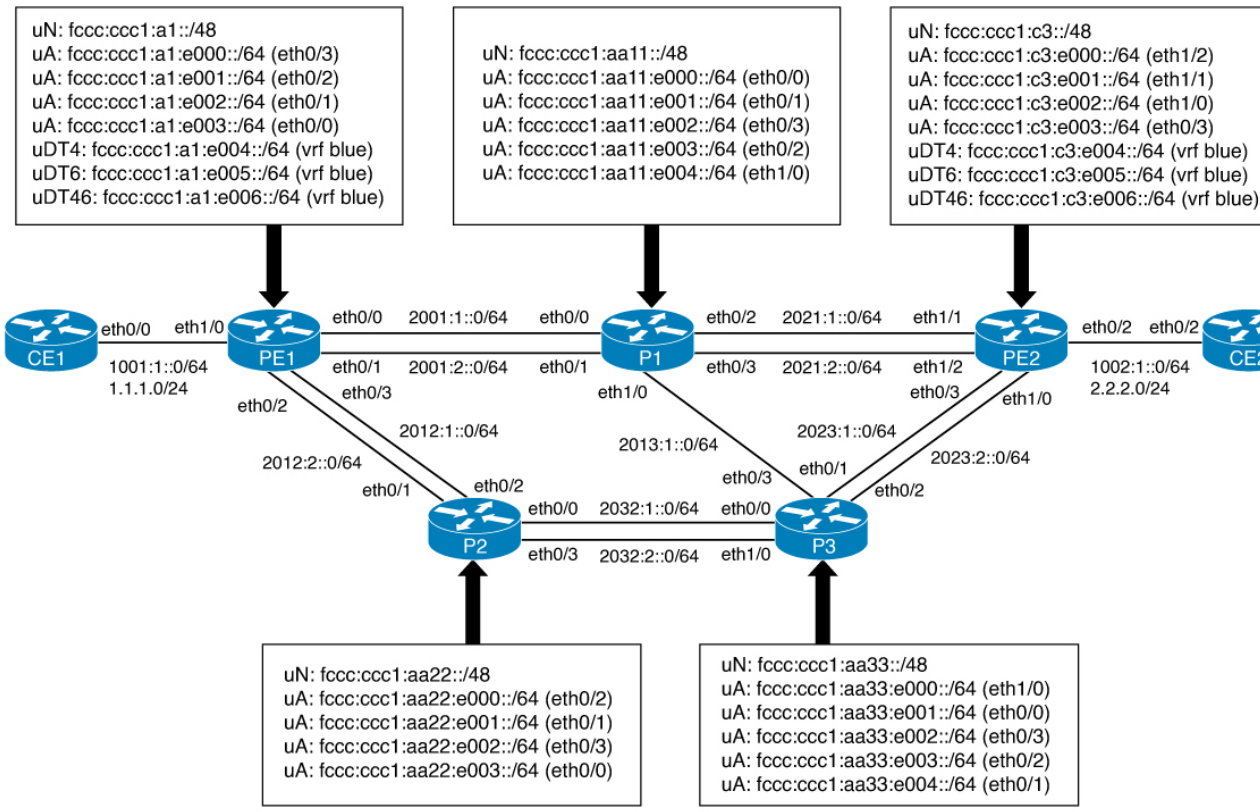
- Traceroute to IPv4 VRF does not display the core PE nodes.
- Ping or Traceroute IPv4 VRF using custom SID-List is not supported.

Information About SRv6 OAM

Operations, Administration, and Maintenance (OAM) helps service providers to monitor SRv6 paths and quickly isolate forwarding problems to assist with fault detection and troubleshooting in the network.

The following figure provides a sample topology for SRv6 OAM.

Figure 44: Sample SRv6 OAM Topology



Operating SRv6 OAM

SRv6 OAM involves the following operations:

- IPv6 Ping/Traceroute CE-CE across SRv6 Core
- IPv4 Ping/Traceroute CE-CE across SRv6 Core
- IPv6 Ping/Traceroute PE-CE across SRv6 Core
- IPv4 Ping/Traceroute PE-CE across SRv6 Core
- IPv6 SID Ping/Traceroute
- IPv6 VRF Ping/Traceroute using custom SIDs

The following examples reference the topology in Figure x.

Operate IPv6 Ping/Traceroute CE-CE across SRv6 Core

Use the following example to operate Ping/Traceroute CE-CE connected IPv6 interface IP:

```
CE1#ping 1002:1::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1002:1::2, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/7 ms
CE1#

CE1#traceroute 1002:1::2 probe 1
Type escape sequence to abort.
Tracing the route to 1002:1::2
  1 1001:1::1 1 msec
  2 1002:1::2 1 msec
CE1#
```

Operate IPv4 Ping/Traceroute CE-CE across SRv6 Core

Use the following example to operate Ping/Traceroute CE-CE connected IPv4 interface IP:

```
CE1#ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
CE1#

CE1#traceroute 2.2.2.2 probe 1
Type escape sequence to abort.
Tracing the route to 2.2.2.2
VRF info: (vrf in name/id, vrf out name/id)
  1 1.1.1.1 1 msec
  2 2.2.2.2 1 msec
CE1#
```

Operate IPv6 Ping/Traceroute PE-CE across SRv6 Core

Use the following example to operate Ping/Traceroute CE's IPv6 interface from PE's VRF interface via SRv6 core:

```
PE1#ping vrf blue 1002:1::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1002:1::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
PE1#

PE1#traceroute vrf blue 1002:1::2 probe 1
Type escape sequence to abort.
Tracing the route to 1002:1::2
  1 2001:1::2 1 msec
  2 2021:2::2 1 msec
  3 1002:1::2 1 msec
PE1#
```

Operate IPv4 Ping/Traceroute PE-CE across SRv6 Core

Use the following example to operate Ping/Traceroute CE's IPv4 interface from PE's VRF interface via SRv6 core:

```
PE1#ping vrf blue 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
PE1#

PE1#traceroute vrf blue 2.2.2.2 probe 1
Type escape sequence to abort.
```

```
Tracing the route to 2.2.2.2
VRF info: (vrf in name/id, vrf out name/id)
  1  *
  2  *
  3 2.2.2.2 1 msec
PE1#
```



Note The IPv4 ping displays "*" instead of IPv6 hops.

Operate IPv6 SID Ping/Traceroute

Use the following example to operate Ping/Traceroute PE2's node SID SRv6 SID from PE1:

```
PE1#ping FCCC:CCC1:C3::
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FCCC:CCC1:C3::, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
PE1#

PE1#traceroute FCCC:CCC1:C3:: probe 1
Type escape sequence to abort.
Tracing the route to FCCC:CCC1:C3::
  1 2001:1::2 0 msec
  2 2021:2::2 0 msec
PE1#
```

Operate IPv6 VRF Ping/Traceroute using Custom SIDs

Use the following example to operate Ping/Traceroute CE2 IPv6 interface from PE1 using custom SRv6 SID list:

The SID in this traceroute goes from PE1 to PE2 via P1, P2 and P3, and finally to CE2.

- The first SID is from PE1 to P1.
- The next SID, from P1 to P2, is an ECMP path via P3 (P1 -> P3 -> P2).
- The next SID, to reach the VPN-SID PE2, is P2 -> P3 -> PE2.
- The last SID is to reach CE2 from PE2 (PE2 -> CE2).

```
PE1#ping srv6 vrf blue 1002:1::2 via segment-list FCCC:CCC1:AA11:AA22:C3:E005::
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1002:1::2 via [
 FCCC:CCC1:AA11:AA22:C3:E005::], timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
PE1#

PE1# traceroute srv6 vrf blue 1002:1::2 via segment-list FCCC:CCC1:AA11:AA22:C3:E005:: probe
 1
Type escape sequence to abort.
Tracing the route to 1002:1::2 via [
 FCCC:CCC1:AA11:AA22:C3:E005::]
  1 2001:1::2 1 msec
  2 2013:1::2 1 msec
  3 2032:2::1 1 msec
  4 2032:1::2 1 msec
```

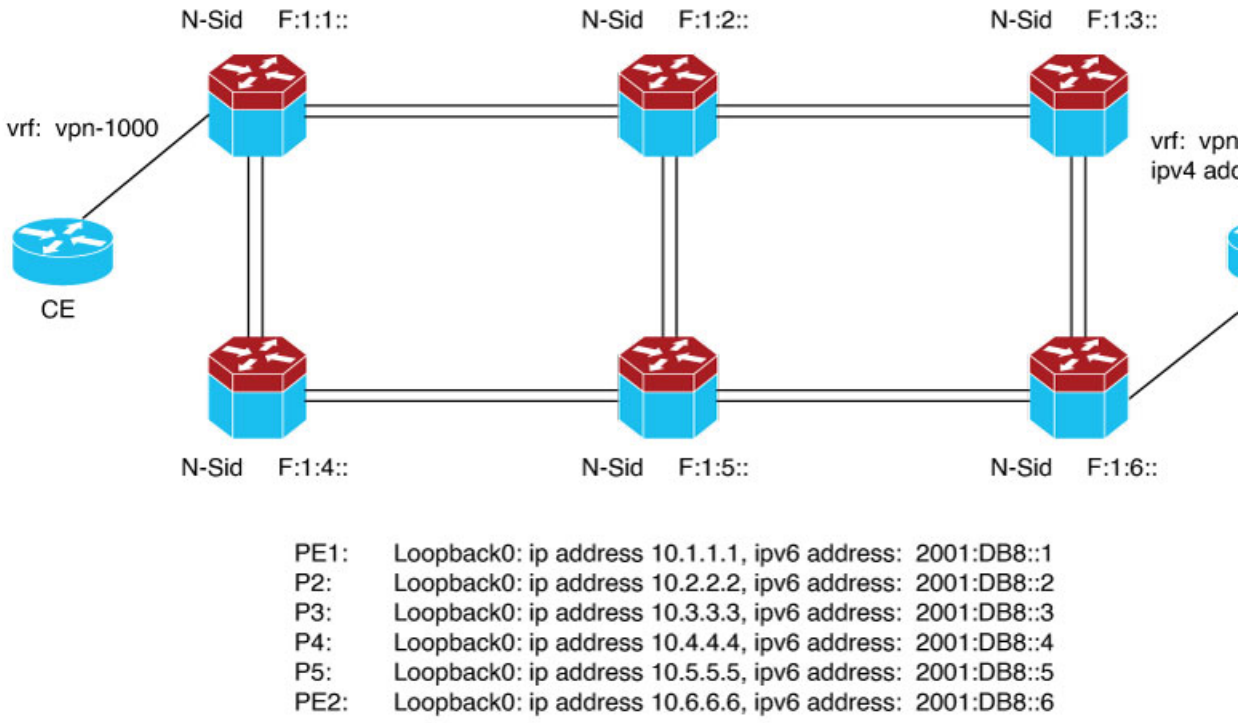
```
  5 2023:1::1 1 msec
  6 1002:1::2 1 msec
PE1#
```

From Cisco IOS XE 17.16.1a, the following operational commands are available:

- `ping srv6 <ipv4-destination> via segment-list […]`
- `traceroute srv6 <ipv4-destination> via segment-list […]`
- `ping srv6 vrf <vrf-name> <ipv4-destination> via segment-list […]`
- `traceroute srv6 vrf <vrf-name> <ipv4-destination> via segment-list […]`
- `traceroute vrf <vrf-name> <ipv4-destination> […]`

The examples below are based on the topology in ***Fig. SRv6 OAM Topology***.

Figure 45: SRv6 OAM Topology




```
PE1#ping srv6 10.6.6.6 via segment-list F:1:4::,F:1:5::,F:1:2::,F:1:3::,F:1:6::
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.6.6.6 via [
 F:1:4::
 F:1:5::
 F:1:2::
 F:1:3::
 F:1:6::], timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms
```

```
PE1#traceroute srv6 10.6.6.6 via segment-list F:1:4::,F:1:5::,F:1:2::,F:1:3::,F:1:6:: probe
 1
Type escape sequence to abort.
```

```
Tracing the route to 10.6.6.6 via [
 F:1:4::
 F:1:5::
 F:1:2::
 F:1:3::
 F:1:6::]
VRF info: (vrf in name/id, vrf out name/id)
  1 1411::4 1 msec
  2 4502::5 1 msec
  3 2511::2 1 msec
  4 2311::3 1 msec
  5 3612::6 1 msec
  6 36.11.0.6 1 msec
```

```
PE1#ping srv6 vrf vpn-1000 10.68.0.6 via segment-list
F:1:4::,F:1:5::,F:1:2::,F:1:3::,F:1:6:E002::
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.68.0.6 via [
 F:1:4::
 F:1:5::
 F:1:2::
 F:1:3::
 F:1:6:E002::], timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
PE1#traceroute srv6 vrf vpn-1000 10.68.0.6 via segment-list
F:1:4::,F:1:5::,F:1:2::,F:1:3::,F:1:6:E002:: probe 1
Type escape sequence to abort.
Tracing the route to 10.68.0.6 via [
 F:1:4::
 F:1:5::
 F:1:2::
 F:1:3::
 F:1:6:E002::]
VRF info: (vrf in name/id, vrf out name/id)
  1 1411::4 0 msec
  2 4502::5 1 msec
  3 2511::2 1 msec
  4 2311::3 1 msec
  5 3612::6 1 msec
  6 68.0.0.6 1 msec
```

```
PE1#traceroute vrf vpn-1000 10.68.0.6 probe 1
Type escape sequence to abort.
Tracing the route to 10.68.0.6
VRF info: (vrf in name/id, vrf out name/id)
  1 1411::4 1 msec
  2 4501::5 1 msec
  3 5611::6 0 msec
  4 10.68.0.6 2 msec
```

Support for SRv6 OAM-TE

From Cisco IOS XE 17.15.1a, the following operational commands provide support for SRv6 OAM traffic engineering:

```
# ping srv6 policy <[<policy-name>] | [color <color-value> endpoint <end-point>]> […]
```

```
# traceroute srv6 policy <[<policy-name>] | [color <color-value> endpoint <end-point>]> […]
```

Note These commands do not support SID list fragmentation.

Use the following examples to operate Ping/Traceroute for OAM traffic engineering:

```
device#ping srv6 policy SRV6PM
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to C02:1::1 via [
 FCCC:CCC1:AA22:AA33:AA11:E001::], timeout is 2 seconds:
Packet sent with a source address of C01:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/9 ms
device#

device#ping srv6 policy color 1 endpoint C02:1::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to C02:1::1 via [FCCC:CCC1:AA22:AA33:AA11:E001::],
timeout is 2 seconds:
Packet sent with a source address of C01:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
device#

device#traceroute srv6 policy SRv6PM
Type escape sequence to abort.
Tracing the route to C02:1::1 via [FCCC:CCC1:AA22:AA33:AA11:E001::]
  1 2012:1::2 1 msec
    2012:2::2 1 msec
    2012:1::2 1 msec
  2 2032:2::2 3 msec 1 msec 3 msec
  3 2013:1::1 0 msec 0 msec 0 msec
  4 2021:2::2 1 msec 1 msec 1 msec
device#

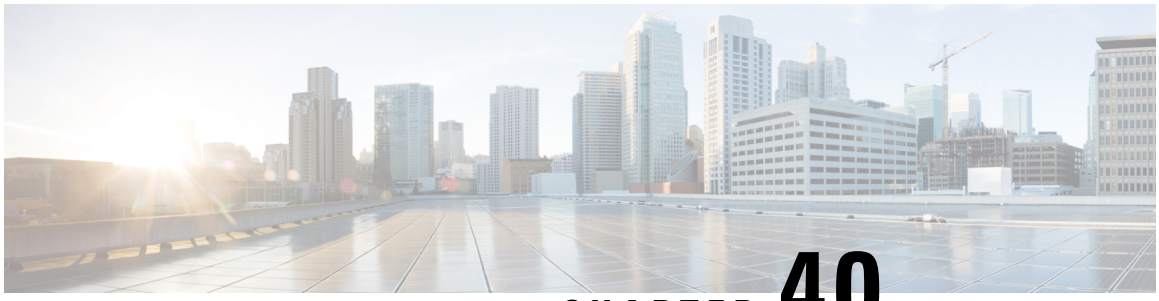
device#traceroute srv6 policy color 1 endpoint C02:1::1
Type escape sequence to abort.
Tracing the route to C02:1::1 via [FCCC:CCC1:AA22:AA33:AA11:E001::]
  1 2012:1::2 1 msec
    2012:2::2 1 msec
    2012:1::2 0 msec
  2 2032:2::2 1 msec 1 msec 0 msec
  3 2013:1::1 1 msec 1 msec 0 msec
  4 2021:2::2 1 msec 1 msec 0 msec
device#
```

Troubleshooting and Debugging SRv6 OAM-TE

Use the following commands to debug SRv6 configuration:

- `# debug platform hardware qfp active feature cef-mpls datapath ipv6 all`
- `# debug srv6 all`
- `# debug isis srv6`
- `# debug bgp ipv6 updates`
- `# debug bgp ipv6 addpath`
- `# debug ip bgp srv6`

- `# debug isis fast-reroute path-selection | ti-lfa | trigger`
- `# debug isis ipv6 microloop`
- `# debug ipv6 packet`



CHAPTER 40

ISIS - SRv6: uLoop Avoidance

From Cisco IOS XE 17.15.1a, the ISIS - SRv6: uLoop Avoidance feature extends the ISIS Local Microloop Protection feature thereby preventing the occurrences of microloops during network convergence after a link-down event or link-up event.

- Prerequisites for ISIS - SRv6: uLoop Avoidance, on page 427
- Restrictions for ISIS - SRv6: uLoop Avoidance, on page 427
- Information About ISIS - SRv6: uLoop Avoidance, on page 427
- How to Enable ISIS - SRv6: uLoop Avoidance, on page 432
- Additional References for ISIS - SRv6: uLoop Avoidance, on page 433
- Feature Information for ISIS - SRv6: uLoop Avoidance, on page 434

Prerequisites for ISIS - SRv6: uLoop Avoidance

- When the Topology-Independent Loop-Free Alternate (TI-LFA) feature is configured, IS-IS SRv6 uloop avoidance is enabled automatically.

Restrictions for ISIS - SRv6: uLoop Avoidance

There are no specific restrictions for this feature.

Information About ISIS - SRv6: uLoop Avoidance

Microloops

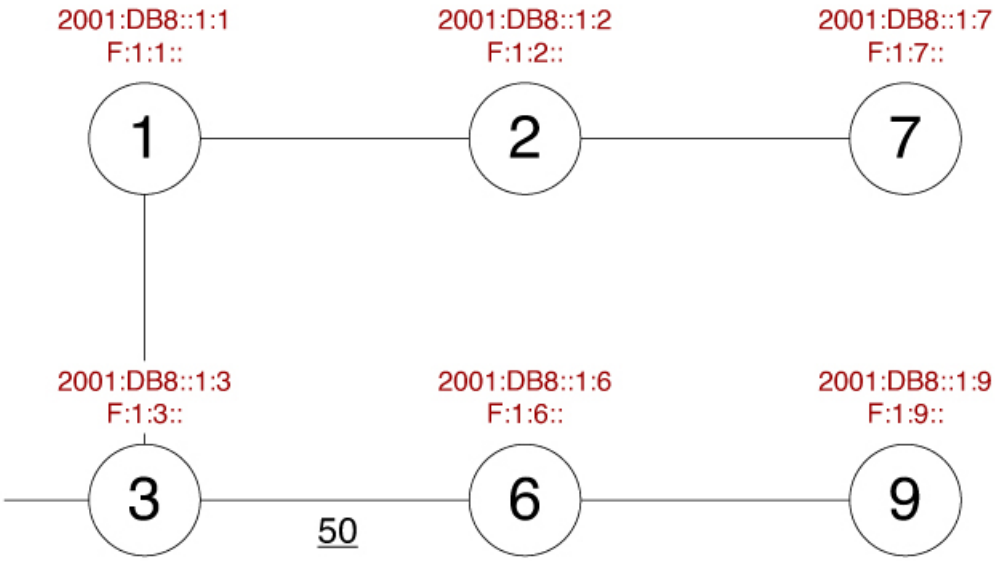
When changes occur in a network topology because of the failure or restoration of a link or a network device, IP Fast Reroute enables rapid network convergence by moving traffic to precomputed backup paths until regular convergence mechanisms move traffic to a newly computed best path, also known as a post-convergence path. This network convergence may cause short microloops between two directly or indirectly connected devices in the topology. Microloops are caused when different nodes in the network calculate alternate paths at different times and independently of each other. For instance, if a node converges and sends traffic to a neighbor node, which has not converged yet, traffic may loop between the two nodes.

Microloops may or may not result in traffic loss. If the duration of a microloop is short, that is the network converges quickly, packets may loop for a short duration before their time-to-live (TTL) expires. Eventually, the packets will get forwarded to the destination. If the duration of the microloop is long, that is one of the routers in the network is slow to converge, packets may expire their TTL or the packet rate may exceed the bandwidth, or the packets might be out of order, and packets may get dropped.

Microloops that are formed between a failed device and its neighbors are called local uloops, whereas microloops that are formed between devices that are multiple hops away are called remote uloops. Local uloops are usually seen in networks where local loop-free alternate (LFA) path is not available. In such networks, remote LFAs provide backup paths for the network.

The information discussed above can be illustrated with the help of an example topology as shown in the following figure.

Figure 46: Microloop Example Topology



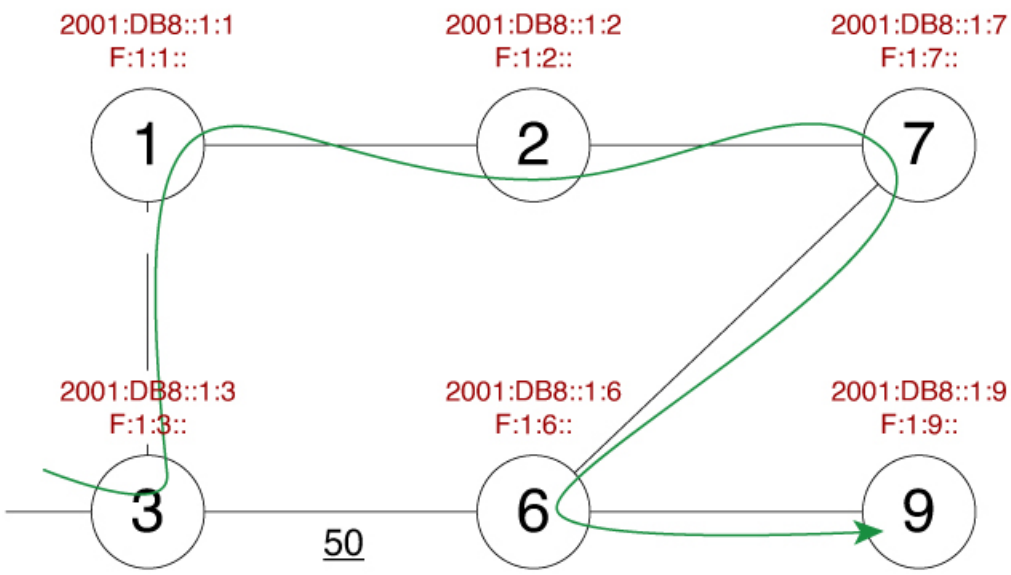
The assumptions in this example are as follows:

- The default metrics is 10 for each link except for the link between Node 3 and Node 6, which has a metric of 50. The order of convergence with SPF backoff delays on each node is as follows:
  - Node 3—50 milliseconds
  - Node 1—500 milliseconds
  - Node 2—1 second
  - Node 7—1.5 seconds

A packet sent from Node 3 to Node 9, the destination, traverses via Node 6.

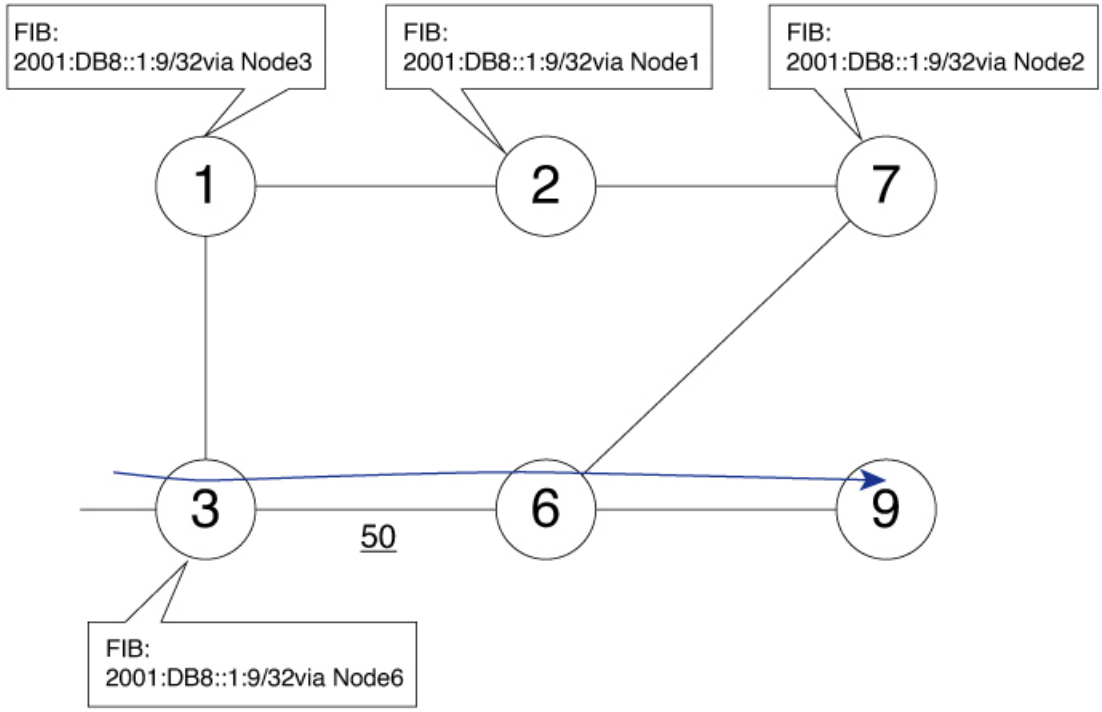
If a link is established between Node 6 and Node 7, the shortest path for a packet from Node 3 to Node 9 would be Node 1, Node 2, Node 7, and Node 6 before the packet reaches the destination, Node 9.

Figure 47: Microloop Example Topology—Shortest Path



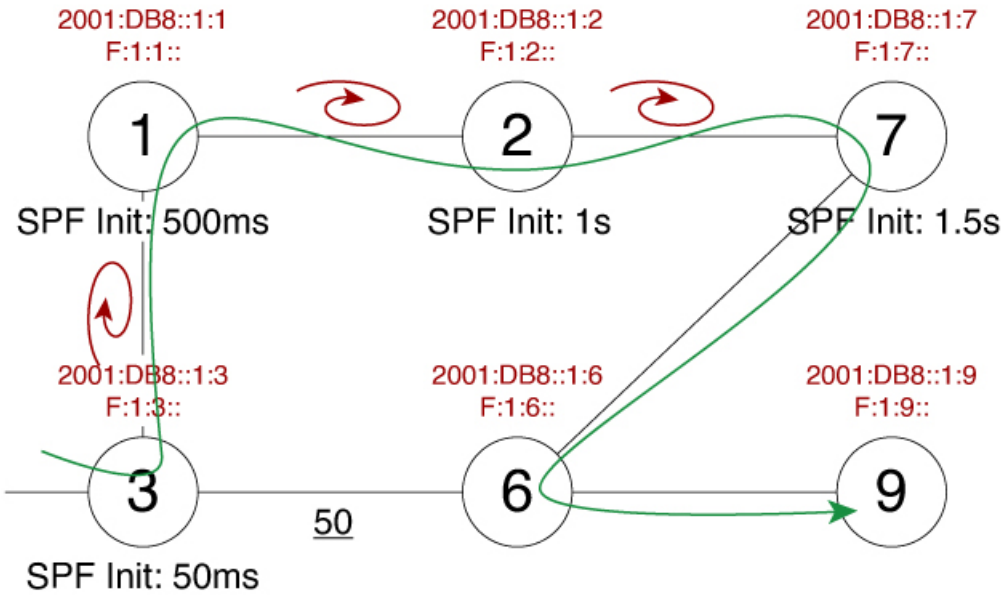
The following figure shows the Forwarding Information Base (FIB) table in each node before the link between Node 6 and Node 7 is established. The FIB entry contains the prefix of the destination node (Node 9) and the next hop.

Figure 48: Microloop Example Topology—FIB Entry



When the link between Node 6 and Node 7 comes up, microloops occur for the links based on the order of convergence of each node. In this example, Node 3 converges first with Node 1 resulting in a microloop between Node 3 and Node 1. Then, Node 1 converges next resulting in a microloop between Node 1 and Node 2. Next, Node 2 converges next resulting in a microloop between Node 2 and Node 7. Finally, Node 7 converges resolving the microloop and the packet reaches the destination Node 9, as shown in the following figure.

Figure 49: Microloop Example Topology—Microloops



Adding the SPF convergence delay, microloop results in a loss of connectivity for 1.5 seconds, which is the convergence duration specified for node 7.

SRv6 and Microloops

The ISIS - SRV6: uLoop Avoidance feature supports the following scenarios, with only one event supported at a time:

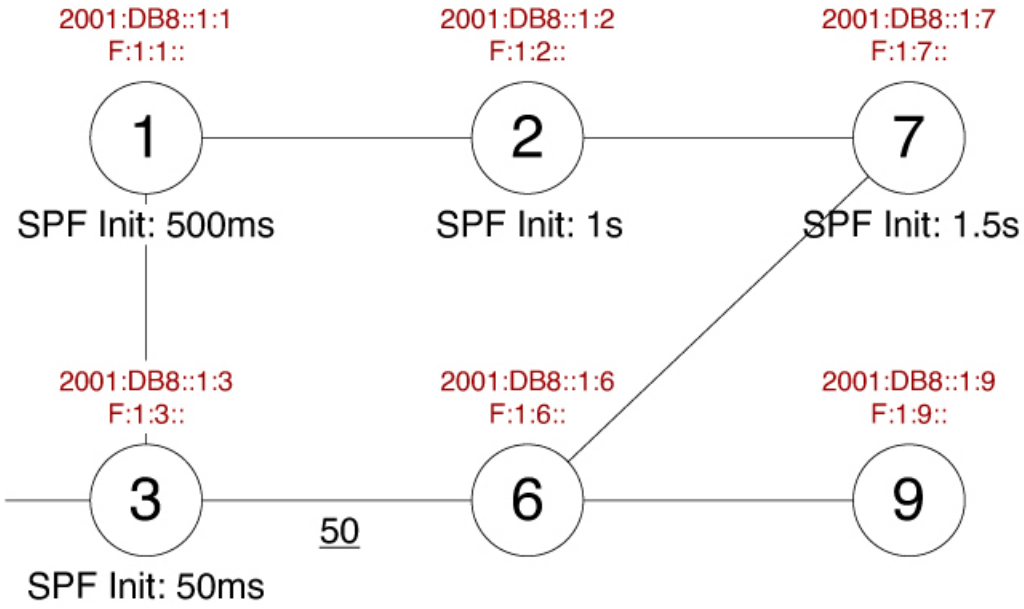
- link-up
- link-down
- link cost increase
- link cost decrease
- overload bit set
- overload bit cleared

Note Node up/down are not supported as these are multiple events.

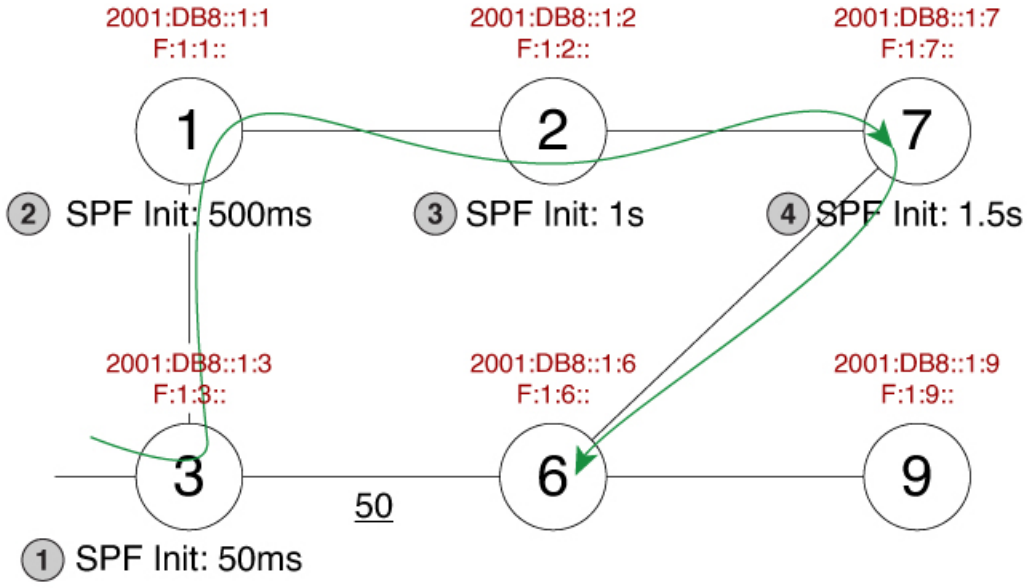
How Segment Routing Prevents Microloops

Using the example used to explain microloops, this section explains how to segment routing prevents microloops.Node 3 in the example is enabled with the **microloop avoidance segment-routing** command. After the link between Node 6 and Node 7 comes up, Node 3 computes a new microloop on the network.

Figure 50: Microloop Example Topology—Segment Routing



Instead of updating the FIB table, Node 3 builds a dynamic loop-free alternate (LFA) SRv6 TE policy for the destination (Node 9) using a list of segments IDs, whichincludes the adjacency segment ID (SID) of Node 6 on Node 7, which is F:1:7:E002::.



So, the SRv6 TE policy enables a packet from Node 3 reaches its destination Node 9, without the risk of microloop until the network converges. Finally, Node 3 updates the FIB for the new path.

Microloop avoidance is available for all IPv6 prefix types, with the **microloop avoidance segment-routing** command, under the **address-family ipv6** configuration. The **microloop avoidance rib-update-delay milliseconds** command can be used to configure the delay in milliseconds for a node to wait before updating the node's forwarding table and stop using the microloop avoidance policy. The default value for the RIB delay is 5000 milliseconds.

Supported Platforms

From Cisco IOS XE 17.15.1a, the ISIS - SRv6: uLoop Avoidance feature is supported on the following platforms:

- Cisco ASR1000 RP3 + ESP100-X, ASR1001-HX, ASR1002-HX
- Cisco Catalyst 8000V Edge Software
- Cisco Catalyst 8200 Series Edge platforms
- Cisco Catalyst 8300 Series Edge platforms
- Cisco Catalyst 8500 and 8500L Series Edge platforms

How to Enable ISIS - SRv6: uLoop Avoidance

Configuring uLoop Avoidance

Use the **microloop avoidance** command under **address-family ipv6** to enable SRv6 uLoop avoidance.

Note SRv6 uLoop avoidance is enabled by default when SRv6 TILFA is enabled.

```
router(config-router-af)# microloop avoidance ?
   disable           Disable Microloop avoidance
   rib-update-delay  Microloop avoidance RIB update delay
   segment-routing   Configuring Segment-Routing Microloop Avoidance
```

The following is a sample configuration to enable SRv6 uLoop avoidance:

```
router isis
address-family ipv6
 microloop avoidance segment-routing
 microloop avoidance rib-update-delay 3000
```

Verifying Microloop Avoidance

Use the **show isis ipv6 rib** command to verify the uLoop configuration.

```
router# show isis ipv6 rib FCCC:CCC1:F1::/48

* FCCC:CCC1:F1::/48
```

```
    via ULOOP_EP SRv6-Fwd-Id 25165888, type L1  metric 120 tag 0
        P node: R5 SID FCCC:CCC1:E1:: uN (PSP/USD)
        Q node: R6 SID FCCC:CCC1:E1:E001:: uA (PSP/USD)
    prefix attr: X:0 R:0 N:0
     (installed)
              - - - - - -
    via FE80::A8BB:CCFF:FE00:9D21/Ethernet1/2, type L1  metric 120 tag 0
    prefix attr: X:0 R:0 N:0
```

Use the **show ipv6 microloop-avoidance** commands to display the current uLoop status.

```
router# show isis ipv6 microloop-avoidance
Tag: 1

Algo  State      Delay        Running(L1/L2)
0     Segment-Routing  3000   FALSE/FALSE
router#
```

```
router# show isis ipv6 microloop-avoidance log

Tag:1 LVL:1
Timestamp Algo Event Reason
*Jul 18 09:49:44.885 CST 0 EXPIRED NA
*Jul 18 09:49:29.885 CST 0 STARTED LINK UP
*Jul 18 09:46:46.064 CST 0 ABORTED NA
*Jul 18 09:46:44.942 CST 0 STARTED LINK DOWN
*Jul 18 09:46:17.536 CST 0 EXPIRED NA
*Jul 18 09:46:02.536 CST 0 STARTED LINK DOWN
*Jul 17 19:39:47.748 CST 0 EXPIRED NA
*Jul 17 19:39:32.748 CST 0 STARTED LINK UP
*Jul 17 19:28:42.973 CST 0 EXPIRED NA
```

Additional References for ISIS - SRv6: uLoop Avoidance

Related Documents

Related Topic	Document Title
Segment Routing and IS-IS	"Segment Routing with IS-IS v4 Node SID" chapter in the *Segment Routing Configuration Guide, Cisco IOS XE 17*
Overview of IS-IS concepts	"IS-IS Overview and Basic Configuration" chapter in the *IP Routing Configuration Guide, Cisco IOS XE 17.x*
ISIS Local Microloop Protection	"ISIS Local Microloop Protection" chapter in the *IP Routing Configuration Guide, Cisco IOS XE 17.x*

Standards/RFCs

Standard/RFC	Title
draft-francois-rtgwg-segment-routing-uloop-00	*Loop avoidance using Segment Routing*

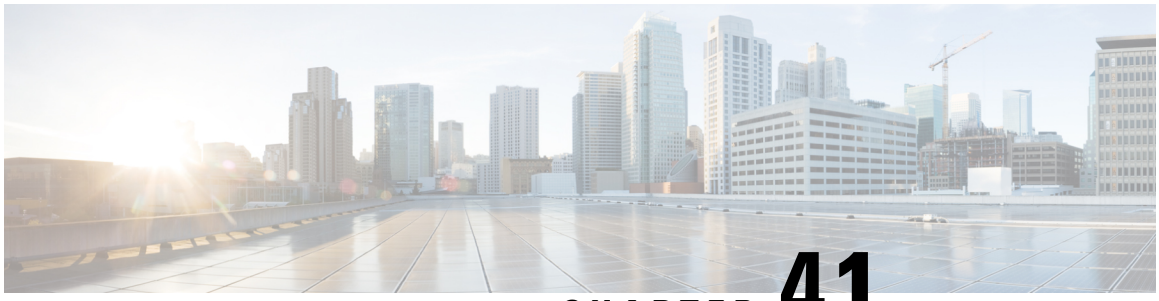
Feature Information for ISIS - SRv6: uLoop Avoidance

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 51: Feature Information for ISIS - SRv6: uLoop Avoidance

Feature Name	Releases	Feature Information
ISIS - SRv6: uLoop Avoidance	Cisco IOS XE 17.15.1a	The ISIS - SRv6: uLoop Avoidance feature extends the ISIS Local Microloop Protection feature to Segment Routing over IPv6 dataplane. This configuration prevents the occurrences of microloops during network convergence after a link-down event or link-up event. The following command was introduced under the **address-family ipv6** sub-mode: **microloop avoidance**



CHAPTER 41

IPv6 Loop-Free Alternate Fast Reroute

When a link or a router fails, distributed routing algorithms compute new routes that take into account the failure. The time taken for this computation is called routing transition. Until the transition is complete and all routers are converged on a common view of the network, the connectivity between the source and destination pairs is interrupted. You can use the IPv6 Loop-Free Alternate (LFA) Fast Reroute (FRR) feature to reduce the routing transition time to less than 50 milliseconds using a precomputed alternate next hop. When a router is notified of a link failure, the router immediately switches over to the repair path to reduce traffic loss.

IPv6 LFA FRR supports the precomputation of repair paths. The repair path computation is done by the Intermediate System-to-Intermediate System (IS-IS) routing protocol, and the resulting repair paths are sent to the IPv6 Routing Information Base (RIB). The repair path installation is done by Cisco Express Forwarding (formerly known as CEF).

- Prerequisites for IPv6 LFA FRR, on page 435
- Restrictions for IPv6 LFA FRR, on page 435
- Information About IPv6 LFA FRR, on page 436
- How to Configure IPv6 LFA FRR, on page 438
- Configuration Examples for IPv6 LFA FRR, on page 441
- Verifying IPv6 LFA FRR Configuration, on page 442
- Feature Information for Configuring IPv6 LFA FRR, on page 442

Prerequisites for IPv6 LFA FRR

There are no specific prerequisites for configuring IPv6 LFA FRR.

Restrictions for IPv6 LFA FRR

- Loop-Free Alternate (LFA) Fast Reroute (FRR) can protect paths that are reachable through an interface only if the interface is a point-to-point interface.
- Any type of tunnel interfaces cannot be used as a protected interface. However, tunnel can be a protecting (repair) tunnel.
- Loadbalance support is available for FRR-protected prefixes on per-prefix basis. If there are multiple equal backup paths, only one can be assigned to a prefix. Assignment is done based on hash function applied to IPv6 prefix. Different IPv6 prefixes have different result of hash function and therefore different backup paths are used.

- A maximum of eight FRR-protected interfaces can simultaneously undergo a cutover.
- Only Layer 3 VPN is supported.
- IPv6 multicast is not supported.
- Only physical and physical port-channel interfaces and subinterfaces are protected. Tunnels and virtual interfaces are not protected.
- The capability of LFA to find a backup path is limited by simplicity of the algorithm. The algorithm can find a backup path only if there is a direct IS-IS neighbor (other than primary one) which has primary path to a prefix, and that primary path does not point to the calculating router. If the network topology is such that LFA cannot cover significant percentage of primary paths with backup paths, it is recommended to use ISIS SRv6 TI-LFA algorithm to get good FRR coverage. For example, LFA algorithm provides good coverage in spine-leaf types of topologies, but not in ring topologies.

Information About IPv6 LFA FRR

IS-IS and IPv6 FRR

When a local link fails in a network, IS-IS recomputes new primary next-hop paths for all affected prefixes. These prefixes are updated in the RIB and the Forwarding Information Base (FIB). Until the primary path prefixes are updated in the forwarding plane, traffic directed towards the affected prefixes are discarded. This process can take hundreds of milliseconds.

In IPv6 FRR, IS-IS computes LFA next-hop routes for the forwarding plane to use in case of primary path failures. LFA is computed per prefix.

When there are multiple LFAs for a given primary path, IS-IS uses a tiebreaking rule to pick a single LFA for a primary path. In case of a primary path with multiple LFA paths, prefixes are distributed equally among LFA paths.

Repair Paths

Repair paths forward traffic during a routing transition. When a link or a router fails, due to the loss of a physical layer signal, initially, only the neighboring routers are aware of the failure. All other routers in the networkare unaware of the nature and location of this failure until information about this failure is propagated through a routing protocol, which may take several hundred milliseconds. It is, therefore, necessary to arrange for packets affected by the network failure to be steered to their destinations.

A router adjacent to the failed link employs a set of repair paths for packets that would have used the failed link. These repair paths are used from the time the router detects the failure until the routing transition is complete. By the time the routing transition is complete, all routers in the network revise their forwarding data and the failed link is eliminated from the routing computation.

Repair paths are precomputed in anticipation of failures so that they can be activated the moment a failure is detected.

The IPv6 LFA FRR feature uses the following repair paths:

- Equal Cost Multipath (ECMP) uses a link as a member of an equal cost path-split set for a destination. The other members of the set can be used as a repair path when the link fails.

- LFA is a next-hop that delivers a packet to its destination without looping back. Downstream paths are a subset of LFAs.

LFA Overview

LFA is a node other than the primary neighbor. Traffic is redirected to an LFA after a network failure. An LFA makes the forwarding decision without any knowledge of the failure.

An LFA must neither use a failed element nor use a protecting node to forward traffic. An LFA must not cause loops. By default, LFA is enabled on all supported interfaces as long as the interface can be used as a primary path.

Advantages of using per-prefix LFAs are as follows:

- The repair path forwards traffic during transition when the primary path link is down.
- All destinations having a per-prefix LFA are protected. This leaves only a subset (a node at the far side of the failure) unprotected.

LFA Calculation

The general algorithms to compute per-prefix LFAs can be found in RFC 5286. IS-IS implements RFC 5286 with a small change to reduce memory usage. Instead of performing a Shortest Path First (SPF) calculation for all neighbors before examining prefixes for protection, IS-IS examines prefixes after SPF calculation is performed for each neighbor. Because IS-IS examines prefixes after SPF calculation is performed, IS-IS retains the best repair path after SPF calculation is performed for each neighbor. IS-IS does not have to save SPF results for all neighbors.

Interaction Between RIB and Routing Protocols

A routing protocol computes repair paths for prefixes by implementing tiebreaking algorithms. The end result of the computation is a set of prefixes with primary paths, where some primary paths are associated with repair paths.

A tiebreaking algorithm considers LFAs that satisfy certain conditions or have certain attributes. When there is more than one LFA, configure the **fast-reroute per-prefix** command with the **tie-break** keyword. If a rule eliminates all candidate LFAs, then the rule is skipped.

A primary path can have multiple LFAs. A routing protocol is required to implement default tiebreaking rules and to allow you to modify these rules. The objective of the tiebreaking algorithm is to eliminate multiple candidate LFAs, select one LFA per primary path per prefix, and distribute the traffic over multiple candidate LFAs when the primary path fails.

Tiebreaking rules cannot eliminate all candidates.

The following attributes are used for tiebreaking:

- Downstream—Eliminates candidates whose metric to the protected destination is lower than the metric of the protecting node to the destination.
- Linecard-disjoint—Eliminates candidates sharing the same linecard with the protected path.
- Shared Risk Link Group (SRLG)—Eliminates candidates that belong to one of the protected path SRLGs.

- Load-sharing—Distributes remaining candidates among prefixes sharing the protected path.
- Lowest-repair-path-metric—Eliminates candidates whose metric to the protected prefix is higher.
- Node protecting—Eliminates candidates that are not node protected.
- Primary-path—Eliminates candidates that are not ECMPs.
- Secondary-path—Eliminates candidates that are ECMPs.

How to Configure IPv6 LFA FRR

Configuring FRR Support



Note LFA computations are enabled for all routes, and FRR is enabled on all supported interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 enable**
5. **ipv6 router isis** *area-tag*
6. **isis network point-to-point**
7. **exit**
8. **router isis** *area-tag*
9. **net** *net*
10. **metric-style wide**
11. **address-family ipv6**
12. **multi-topology**
13. **fast-reroute per-prefix** {**level-1 | level-2**} {**all | route-map** *route-map name*}
14. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 on the interface. You can also enable IPv6 by configuring an IPv6 address.
Step 5	**ipv6 router isis** *area-tag* **Example:** `Device(config-if)# ipv6 router isis ipfrr`	Configures an IS-IS routing process for an IPv6 on an interface and attaches an area designator to the routing process.
Step 6	**isis network point-to-point** **Example:** `Device(config-if)# isis network point-to-point`	Enforces IS-IS point-to-point network type.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**router isis** *area-tag* **Example:** `Device(config)# router isis ipfrr`	Enables the IS-IS routing protocol, specifies an IS-IS process, and enters router configuration mode.
Step 9	**net** *net* **Example:** `Device(config-router)# net 49.0001.0101.2800.0001.00`	Configures an IS-IS network entity (NET) for a routing process.
Step 10	**metric-style wide** **Example:** `Device(config-router)# metric-style wide`	Enables metric-style wide. **Note** It is recommended to run wide metric o
Step 11	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Enters IPv6 configuration sub-mode.

	Command or Action	Purpose
Step 12	**multi-topology** **Example:** `Device(config-router-af)# multi-topology`	(Optional) Allows IS-IS to run in multi-topology mode in compliance with RFC 5120. Multi-topology allows for non-concurrent IPv4 and IPv6 topologies. **Note** IS-IS supports IPv6 LFA also in single-top
Step 13	**fast-reroute per-prefix** {**level-1** | **level-2**} {**all** | **route-map** *route-map name*} **Example:** `Device(config-router-af)# fast-reroute per-prefix level-2 all`	Enables per-prefix FRR in LFA mode. **Note** Configure the **all** keyword to protect all pre
Step 14	**end** **Example:** `Device(config-router-af)# end`	Exits router configuration mode and enters privileged EXEC mode.

Additional IS-IS IPv6 Commands

From Cisco IOS XE 17.15.1a, you can use the following optional commands to further fine-tune LFA FRR configurations:

Router IS-IS / Address-family IPv6 Mode Commands

fast-reroute tie-break {level-1 | level-2}

Configures the following tie-breakers that impact backup path calculation and selection:

```
downstream                      Prefer repair path via downstream node
linecard-disjoint               Prefer line card disjoint repair path
lowest-backup-path-metric       Prefer repair path with lowest total metric
node-protecting                 Prefer node protecting repair path
primary-path                    Prefer repair path from ECMP set
secondary-path                  Prefer non-ECMP repair path
srlg-disjoint                   Prefer SRLG disjoint repair path
```

fast-reroute interface disable <level>

Disables FRR protection on all interfaces by default. Interfaces where FRR is required can be configured explicitly using the interface level command.

fast-reroute load-sharing <level> disable

Disables load sharing between equal backup paths.

fast-reroute use-candidate-only <level>

Use as candidate interface only these allowed by the interface configuration.

Interface IS-IS IPv6 FRR Commands

isis ipv6 fast-reroute candidate <level> {disable}

Configures the interface for fast-reroute backup path.

isis ipv6 fast-reroute exclude <level> <interface>

Excludes another interface from being used for fast-reroute backup.

isis ipv6 fast-reroute protection <level> {disable}

Enables or disables fast-reroute protection on an interface.

isis ipv6 fast-reroute tie-break <level>

Creates the following set of tie-breakers specific for the interface:

```
default                          Use default tiebreakers set
downstream                       Prefer repair path via downstream node
linecard-disjoint                Prefer line card disjoint repair path
lowest-backup-path-metric        Prefer repair path with lowest total metric
node-protecting                  Prefer node protecting repair path
primary-path                     Prefer repair path from ECMP set
secondary-path                   Prefer non-ECMP repair path
srlg-disjoint                    Prefer SRLG disjoint repair path
```

Configuration Examples for IPv6 LFA FRR

Example: Configuring IPv6 LFA FRR

The following example shows basic configuration of IPv6 LFA FRR on the router interface and under router ISIS. IPv6 LFA FRR is enabled in level 2 for all ISIS IPv6 prefixes present in level 2.

```
interface Ethernet0/0
 ip unnumbered Loopback0
 ipv6 enable
  ipv6 router isis 1
   isis network point-to-point
 !
router isis 1
 net 49.0000.2222.2222.2222.00
 is-type level-2-only
 router-id Loopback0
  metric-style wide

 address-family ipv6
  multi-topology
  router-id Loopback0
   fast-reroute per-prefix level-2 all
exit-address-family
```

In the following example, only routes with tag 17 are protected.

```
router isis
 net 47.0004.004d.0001.0001.c11.1111.00
 address-family ipv6
  fast-reroute per-prefix level-2 route-map ipfrr-include
exit
route-map ipfrr-include
  match tag 17
```

Verifying IPv6 LFA FRR Configuration

Use the following show commands to verify IPv6 FRR and LFA configuration:

show isis ipv6 fast-reroute interfaces

```
router# show isis ipv6 fast-reroute interfaces

Tag 1 - Fast-Reroute Platform Support Information:

 SRv6 TI-LFA: Supported by platform
 Level-1 MT-2: FRR: Not Enabled, TI-LFA: Not Enabled
 Level-2 MT-2: FRR: Enabled, TI-LFA: Not Enabled
  Ethernet1/3: Protectable: Yes. Usable for repair: Yes
  Ethernet1/1: Protectable: Yes. Usable for repair: Yes
  Ethernet1/0: Protectable: Yes. Usable for repair: Yes
```

show isis ipv6 fast-reroute summary

```
router# show isis ipv6 fast-reroute sum

Tag 1:
IPv6 Fast-Reroute Protection Summary:

 Prefix Counts:          Total      Protected    Coverage
   High priority:        0          0            0%
   Normal priority:      12         3            25%
   Total:                12         3            25%
```

show isis ipv6 rib

```
router# show isis ipv6 rib 604::1/128

IS-IS IPv6 process 1, local RIB

Repair path attributes:
    DS - Downstream, LC - Linecard-Disjoint, NP - Node-Protecting
    PP - Primary-Path, SR - SRLG-Disjoint

* 604::1/128  prefix attr X:0 R:0 N:1
    via FE80::A8BB:CCFF:FE02:5E20/Ethernet0/2, type L2  metric 40 tag 0
    prefix attr: X:0 R:0 N:1
     (installed)
     repair path: via FE80::A8BB:CCFF:FE02:5A00/Ethernet0/0 metric: 40 (PP,DS,NP,SR)
      local LFA
      repair source: r604, metric to pfx: 40
    via FE80::A8BB:CCFF:FE02:5A00/Ethernet0/0, type L2  metric 40 tag 0
    prefix attr: X:0 R:0 N:1
     (installed)
     repair path: via FE80::A8BB:CCFF:FE02:5E20/Ethernet0/2 metric: 40 (PP,DS,NP,SR)
      local LFA
      repair source: r604, metric to pfx: 40
```

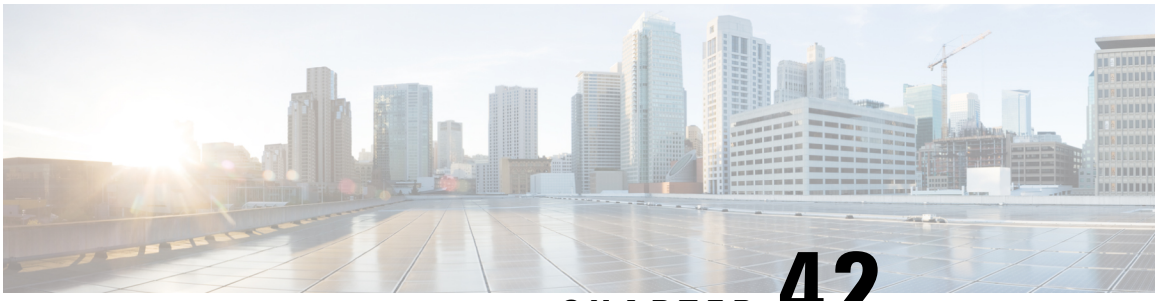
Feature Information for Configuring IPv6 LFA FRR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 52: Feature Information for Configuring IPv6 LFA FRR

Feature Name	Releases	Feature Information
IPv6 Loop-Free Alternate Fast Reroute	Cisco IOS XE Release 17.15.1a	This feature was introduced. The following commands are introduced or modified as part of this feature: **fast-reroute tie-break {level-1 | level-2}** **isis ipv6 fast-reroute candidate <level> {disable}**



CHAPTER 42

IS-IS SRv6 Link-protection Topology Independent Loop Free Alternate Fast Reroute

This document describes the functionalities and IS-IS implementation of IPv6 Fast Re-Route feature (IPv6FRR) using Segment Routing over IPv6 (SRv6) Topology Independent Loop Free Alternative (TI-LFA) link protection.

- Feature Information for IS-IS SRv6 Link-protection TI-LFA FRR, on page 445
- Prerequisites for IS-IS SRv6 Link-protection TI-LFA FRR , on page 445
- Restrictions for IS-IS SRv6 Link-protection TI-LFA FRR, on page 446
- Information About IS-IS SRv6 Link-protection TI-LFA FRR, on page 446
- How to Configure IS-IS SRv6 Link-protection TI-LFA FRR , on page 448

Feature Information for IS-IS SRv6 Link-protection TI-LFA FRR

Table 53: Feature Information for IS-IS SRv6 Link-protection TI-LFA FRR

Feature Name	Releases	Feature Information
IS-IS SRv6 Link-protection Topology Independent Loop Free Alternate Fast Reroute	Cisco IOS XE 17.15.1a	This feature was introduced.
IPv6 GRE-TP tunnel as protected link support for SRv6 TI-LFA with ISIS	Cisco IOS XE 17.18.2	From Cisco IOS XE 17.18.2, this feature extends IPv6 GRE-TP tunnel as protected link support for SRv6 TILFA with ISIS.

Prerequisites for IS-IS SRv6 Link-protection TI-LFA FRR

- SRv6 must be enabled on all the nodes, before configuring SRv6 TI-LFA. To enable SRv6, see chapter Segment Routing over IPv6.

```
segment-routing srv6
 locators
  locator A
   prefix CAFE:0:601::/48
   format usid-f3216
```

```
  !
router isis 1
net 49.0000.1111.1111.1111.00
 is-type level-2-only
 router-id Loopback0
 metric-style wide
address-family ipv6
  multi-topology
  router-id Loopback0
  segment-routing srv6
   locator A
    level-2
exit-address-family
!
```

Restrictions for IS-IS SRv6 Link-protection TI-LFA FRR

- The IGP cannot guarantee loop-free paths when tunnels are used for protection or repair, as it is unaware of the tunnel's underlying physical path.
- Forwarding failures on tunnel interfaces can occur in cases where local link failures do not occur. For this reason, in order to quickly detect forwarding failures on tunnel interfaces it is advisable to use the BFD protocol.

  Note that aggressive BFD timers must be used with care on tunnel interfaces as they can lead to false indication of failures
- Due to limitations on this platform, IGP TI-LFA cannot guarantee 50ms FRR failover.

Information About IS-IS SRv6 Link-protection TI-LFA FRR

When the local LFA is enabled, there is not always a good coverage of the prefixes to be protected.

To overcome the above limitation, effective Cisco IOS-XE Release 17.15.1a, topology-independent LFA (TI-LFA) is supported on an SRv6-enabled network.

Topology-Independent Loop Free Alternate

TI-LFA provides supports for the following:

- Link Protection—The LFA provides repair path for failure of the link.
- Local LFA—Whenever a local LFA on the post convergence path is available, it is preferred over TI-LFA because local LFA does not require additional SID for the repair path. That is, the label for the PQ node is not needed for the release node.
- TI-LFA path to traverse multiple intersect or disjoint PQ nodes, up to the platform’s maximum supported labels—TI-LFA provides complete coverage of all prefixes.
- P2P interfaces for the protected link—TI-LFA protects P2P interfaces.
- Asymmetrical links—The ISIS metrics between the neighbors are not the same.
- Multi-homed (anycast) prefix protection—The same prefix may be originated by multiple nodes.

- Protected prefix filtering—The route-map includes or excludes a list of prefixes to be protected and the option to limit the maximum repair distance to the release node.
- Tiebreakers—A subset of existing tiebreakers, applicable to TI-LFA, is supported.

Topology Independent Loop Free Alternate Tie-break

Local and remote LFA use default or user-configured heuristics to break the tie when there is more than one path to protect the prefix. The attributes are used to trim down the number of repair paths at the end of the TI-LFA link protection computation before the load balancing. Local LFA and remote LFA support the following tiebreakers:

- Linecard-disjoint—Prefers the line card disjoint repair path
- Lowest-backup-path-metric—Prefers the repair path with lowest total metric
- Node-protecting—Prefers node protecting repair path
- SRLG-disjoint—Prefers SRLG disjoint repair path
- Load-sharing—Distributes repair paths equally among links and prefixes

When there are two repair paths for a particular prefix, the path that the output port on different line card than that of the primary port is chosen as the repair path. For TI-LFA link protection, the following tiebreakers are supported:

- Linecard-disjoint—Prefers the line card disjoint repair path.
- LC disjoint index—If both the repair paths are on the same line card as that of the primary path, then, both paths are considered as candidates. If one of the path is on a different line card, then that path is chosen as the repair path.
- SRLG index—If both the repair paths have the same SRLG ID as that of the primary path, then, both the paths are considered as candidates. If one of the path has a different srlg id, then path is chosen as the repair path.
- Node-protecting—For TI-LFA node protection, the protected node is removed when computing the post-convergence shortest path. The repair path must direct traffic around the protected node.

The SRLG ID can be configured for each interface. When there are two repair paths for a prefix, the configured SRLG ID for the repair path is compared with that of the primary path SRLG ID. If the SRLG IDs for the secondary path is different than that of the primary, that path is chosen as the repair path. This policy comes into effect only when the primary path is configured with an SRLG ID. It is possible to configure both node and SRLG protection modes for the same interface or the same protocol instance. In that case, an additional TI-LFA node-SRLG combination protection algorithm is run. The TI-LFA node-SRLG combination algorithm removes the protected node and all members of the interface with the same SRLG group when computing the post-convergence SPT.

Interface Fast Reroute Tiebreakers

Interface fast reroute (FRR) tiebreakers are also needed for TI-LFA node and SRLG protection. When interface and protocol instance FRR tiebreakers both are configured, the interface FRR tiebreakers take precedence over the protocol instance. When interface FRR tiebreakers are not configured, the interface inherits the protocol instance FRR tiebreakers.

The following interface FRR tiebreaker commands apply only to the particular interface.

```
isis fast-reroute tie-break
[level-1 | level-2] linecard-disjoint
priority
isis fast-reroute tie-break
[level-1 | level-2] lowest-backup-metric
priority
isis fast-reroute tie-break
[level-1 | level-2] node-protecting
priority
isis fast-reroute tie-break
[level-1 | level-2] srlg-disjoint
priority
isis fast-reroute tie-break
[level-1 | level-2] default
```

Tie-breaker default and explicit tie-breaker on the same interface are mutually exclusive.

The following tie-breakers are enabled by default on all LFAs:

- linecard-disjoint
- lowest-backup-metric
- srlg-disjoint

How to Configure IS-IS SRv6 Link-protection TI-LFA FRR

Perform the following steps to configure SRv6 Link-protection Topology Independent Loop Free Alternate Fast Reroute.

Configuring Topology Independent Loop Free Alternate Fast Reroute

You can enable TI-LFA using any of the following two methods:

1. **Protocol enablement**—Enables TI-LFA in router isis mode for all IS-IS interfaces. Optionally, use the interface command to exclude the interfaces on which TI-LFA should be disabled.

   For example, to enable TI-LFA for all IS-IS interfaces:

   ```
   router isis 1
    address-family ipv6
      fast-reroute per-prefix {level-1 | level-2} {all | route-map route-map-name}
      fast-reroute ti-lfa {level-1 | level-2} [maximum-metric value]
   ```



Note The **isis fast-reroute per-prefix level-x** command enables local LFA and is required to enable TI-LFA.

2. **Interface enablement**—Enable TI-LFA selectively on each interface.

   ```
   interface interface-name
     isis fast-reroute protection {level-1 | level-2}
     isis fast-reroute ti-lfa protection {level-1 | level-2} [maximum-metric value]
   ```

The **maximum-metric** option specifies the maximum repair distance which a node is still considered eligible as a release node.

When both interface and protocol are TI-LFA enabled, the interface configuration takes precedence over the protocol configuration. TI-LFA is disabled by default.

To disable TI-LFA on a particular interface, use the following command:

```
interface interface-name
  isis fast-reroute ti-lfa protection {level-1 | level-2} disable
```

Examples: Configuring IS-IS SRv6 Link-protection TI-LFA FRR

Example 1: Base IS-IS SRv6 TILFA configuration

```
segment-routing srv6
 locators
  locator A
   prefix CAFE:0:601::/48
   format usid-f3216
  !
router isis 1
net 49.0000.1111.1111.1111.00
 is-type level-2-only
 router-id Loopback0
 metric-style wide
address-family ipv6
  router-id Loopback0
  multi-topology
  segment-routing srv6
   locator A
  level-2
  fast-reroute per-prefix level-2 all
  fast-reroute ti-lfa level-2
exit-address-family
!
```

Example 2: EnableTI-LFA node-protecting tie-breaker on all IS-IS level-2 interfaces with priority 100. All other tie-breakers are disabled.

```
router isis
 address-family ipv6

  fast-reroute per-prefix level-2 all
  fast-reroute ti-lfa level-2
  fast-reroute tie-break level-2 node-protecting 100
```

Example 3: Enable TI-LFA node-protecting tie-breaker with priority 100 and TI-LFA SRLG protection with priority 200 on all IS-IS level-2 interfaces. All other tie-breakers are disabled because the node-protecting tie-breaker is configured.

```
router isis
 address-family ipv6

   fast-reroute per-prefix level-2 all
   fast-reroute ti-lfa level-2
   fast-reroute tie-break level-2 node-protecting 100
   fast-reroute tie-break level-2 srlg-disjoint 200
```

Example 4: Enable TI-LFA node-protecting tie-breaker with priority 100 on all IS-IS level-2 interfaces except on Ethernet0/0. For those IS-IS interfaces, all other tie-breakers are disabled. Ethernet0/0 overwrites the inheritance and uses the default set of tie-breakers with linecard-disjoint, lowest-backup-path-metric, srlg-disjoint enabled.

```
router isis
address-family ipv6
 fast-reroute per-prefix level-2 all
 fast-reroute ti-lfa level-2
 fast-reroute tie-break level-2 node-protecting 100
!
 interface ethernet0/0
  ipv6 router isis
  isis ipv6 fast-reroute tie-break level-2 default
```

Example 5: Enable TI-LFA using the default tie-breaker on all IS-IS interfaces except on Ethernet0/0. On Ethernet0/0, enable TI-LFA node-protecting with priority 100 and disable all other tie-breakers.

```
router isis
address-family ipv6
 fast-reroute per-prefix level-2 all
 fast-reroute ti-lfa level-2
!
 interface ethernet0/0
  ipv6 router isis
  isis ipv6 fast-reroute tie-break level-2 node-protecting 100
```

Example 6: Enable TI-LFA node-protecting tie-breaker with priority 200 and linecard-disjoint tie-breaker with priority 100 on all IS-IS level-2 interfaces. All other tie-breakers are disabled.

```
router isis
address-family ipv6

 fast-reroute per-prefix level-2 all
 fast-reroute ti-lfa level-2
 fast-reroute tie-break level-2 linecard-disjoint 100
 fast-reroute tie-break level-2 node-protecting 200
```

Verifying the Tiebreaker

To view tiebreakers enabled on the interface, use the following command:

show running all | section interface *interface-name*

To view tiebreakers enabled on the router mode, use the following command:

show running all | section router isis

Verifying the Primary and Repair Paths

Use the following show commands to verify primary and repair paths:

```
router#show isis ipv6 rib 605::1/128

IS-IS IPv6 process 1, local RIB

Repair path attributes:
```

```
    DS - Downstream, LC - Linecard-Disjoint, NP - Node-Protecting
    PP - Primary-Path, SR - SRLG-Disjoint

* 605::1/128  prefix attr X:0 R:0 N:1
    via FE80::A8BB:CCFF:FE02:5E20/Ethernet0/2, type L2  metric 30 tag 0
    prefix attr: X:0 R:0 N:1
     (installed)
     repair path: via FE80::A8BB:CCFF:FE02:5A00/Ethernet0/0 metric: 30 (DS,NP)
      TI-LFA node-protecting, link-protecting
        SRv6-Fwd-Id 25165857
        P node: r604 SID CAFE:0:604:: uN (PSP/USD)
      repair source: r605, metric to pfx: 60

router#show ipv6 route 605::1/128
Routing entry for 605::1/128
  Known via "isis 1", distance 115, metric 30, type level-2
  Route count is 1/1, share count 0
  Routing paths:
    FE80::A8BB:CCFF:FE02:5E20, Ethernet0/2 [Primary, repair via " Bind Label: 25165857"]
      Route metric is 30, traffic share count is 1
      From FE80::A8BB:CCFF:FE02:5E20
      Last updated 00:59:08 ago
     Bind Label: 25165857 [Repair]
      Route metric is 30, traffic share count is 1
      Last updated 00:59:08 ago

router#show ipv6 cef 605::1/128
605::1/128
  nexthop FE80::A8BB:CCFF:FE02:5E20 Ethernet0/2
    repair: recursive 25165857


router#show ipv6 cef 605::1/128 internal
605::1/128, epoch 0, RIB[I], refcnt 4, per-destination sharing
  sources: RIB
  feature space:
    IPRM: 0x00028000
  ifnums:
    Ethernet0/2(4): FE80::A8BB:CCFF:FE02:5E20
  path list 7F514ADDD0E0, 43 locks, per-destination, flags 0x16D [shble, hvsh, rif, rcrsv,
 hwcn, bldmp]
    path 7F514ABD30B8, share 1/1, type attached nexthop, for IPv6, flags [has-rpr]
      nexthop FE80::A8BB:CCFF:FE02:5E20 Ethernet0/2, IPV6 adj out of Ethernet0/2, addr
FE80::A8BB:CCFF:FE02:5E20 7F514B05AA48
        repair: recursive 25165857[Binding-Label:Default] (7F514ABD3188)
    path 7F514ABD3188, share 1/1, type recursive, for IPv6, flags [rpr, rpr-only]
      recursive via 25165857[Binding-Label:Default], repair, fib 7F514B624830, 1 terminal
fib, blbl:Default:25165857
      path list 7F514ADDD2F0, 3 locks, per-destination, flags 0x28049 [shble, rif, hwcn,
sb-oce, spec-order]
          path 7F514ABD34C8, share 1/1, type attached nexthop, for IPv6
            nexthop FE80::A8BB:CCFF:FE02:5A00 Ethernet0/0, IPV6 adj out of Ethernet0/0,
addr FE80::A8BB:CCFF:FE02:5A00 7F5145297A28
  output chain:
    loadinfo 7F514B15E908, per-session, 1 choice, flags 0185, 25 locks
      flags [Per-session, for-rx-IPv6, 2buckets, indirection]
      1 hash bucket
        < 0 > FRR Primary (0x7F514B15D9B8)
                <primary: IPV6 adj out of Ethernet0/2, addr FE80::A8BB:CCFF:FE02:5E20
7F514B05AA48>
               <repair:  SRv6 SID List OCE 0x7F514B05D4D8 (7) 1 Segments, Flags 0x0 [none]

                          Segment List (1) mode:insert
                            CAFE:0:604:: [SL-MSD:16 END-POP-MSD:16 SRH-Inst:1]
                         PushCounter(SRv6 Encap) 7F514489F2E0
```

```
                              SRv6 Encap OCE 0x7F514B05FC68 (4) fwd-id:25165857 CAFE:0:604::
                                Encap Flags: 00000000 [none]
                                IPv6 TC: 0   Flow Label: 0       Hop Limit: 64
                                  Src: 601::1
                                  Dst: CAFE:0:604::
                              IPV6 adj out of Ethernet0/0, addr FE80::A8BB:CCFF:FE02:5A00
7F5145297A28>
      Subblocks:  None
```

Verifying SRv6 Configuration

Use the **show segment-routing srv6 locator** command to view SRv6 locators:

```
router# show segment-routing srv6 locator
  Name                  ID    Algo    Prefix             Status    Flags
  myLoc1                 3      0      2001:0:8::/48        Up        U
  myLocBestEffort        5      0      2001:0:1::/48        Up        U
```

Use the **show isis srv6 locators** command to view SID locators:

```
router# show isis srv6 locators
ISIS SRv6 Locators:
Tag sr:
Name             Prefix                       Level
----             ------                       -----
loc1             FC01:101:2::/48              2
```

```
router# show isis srv6 locators detail
ISIS SRv6 Locators:
Tag sr:
Name             Prefix                        Level
----             ------                        -----
loc1             FC01:101:2::/48               2

Level-1 metric: 0
 Level-2 metric: 0
 End-SIDs:
   FC01:101:2::
```

Verifying the IS-IS Topology Independent Loop Free Alternate Paths

Use the following show commands to verify ISIS TI-LFA configuration:

```
router#show isis ipv6 fast-reroute ti-lfa fwd-ids

Tag 1:

SRv6-Fwd-Id: 25165858 via FE80::A8BB:CCFF:FE02:5A00 Ethernet0/0 (2)
  Uncompressed SID List, SID count: 1
    P node: r603.00-00 SID CAFE:0:603:: uN (PSP/USD)
  Compressed SID List, SID count: 1
    SID[1]: CAFE:0:603::, MSD SL:16 End-POP:16

SRv6-Fwd-Id: 25165857 via FE80::A8BB:CCFF:FE02:5A00 Ethernet0/0 (2)
  Uncompressed SID List, SID count: 1
    P node: r604.00-00 SID CAFE:0:604:: uN (PSP/USD)
  Compressed SID List, SID count: 1
    SID[1]: CAFE:0:604::, MSD SL:16 End-POP:16

SRv6-Fwd-Id: 25165856 via FE80::A8BB:CCFF:FE02:5E20 Ethernet0/2 (4)
  Uncompressed SID List, SID count: 1
    P node: r604.00-00 SID CAFE:0:604:: uN (PSP/USD)
```

```
  Compressed SID List, SID count: 1
    SID[1]: CAFE:0:604::, MSD SL:16 End-POP:16

router#
```

```
router# show isis ipv6 fast-reroute summary

Tag 1:
IPv6 Fast-Reroute Protection Summary:

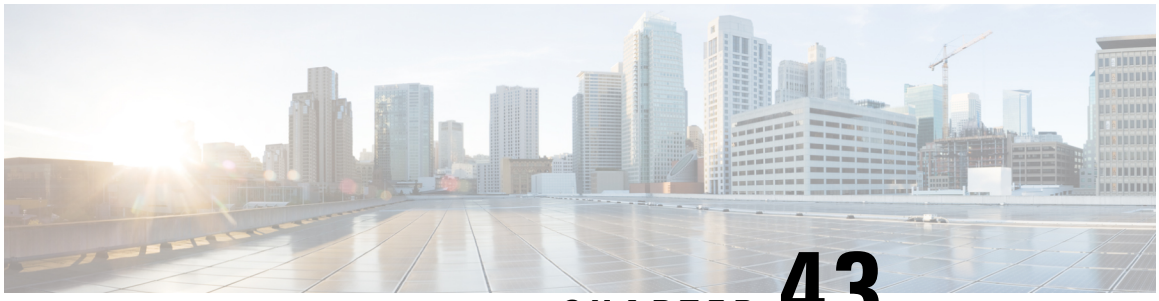
Prefix Counts: Total Protected Coverage
High priority: 0 0 0%
Normal priority: 25 22 88%
Total: 25 22 88%
```

```
router# show isis ipv6 fast-reroute interfaces


Tag 1 - Fast-Reroute Platform Support Information:

SRv6 TI-LFA: Supported by platform
Level-1 MT-2: FRR: Enabled, TI-LFA: Enabled
Level-2 MT-2: FRR: Not Enabled, TI-LFA: Not Enabled

TenGigabitEthernet2/3/0.3: Protectable: Yes. Usable for repair: Yes
GigabitEthernet2/1/1.2: Protectable: Yes. Usable for repair: Yes
GigabitEthernet2/1/1.1: Protectable: Yes. Usable for repair: Yes
Tunnel122: Protectable: Yes. Usable for repair: Yes
Tunnel121: Protectable: Yes. Usable for repair: Yes
Tunnel16: Protectable: Yes. Usable for repair: Yes
Loopback0: Protectable: Yes. Usable for repair: Yes
```



CHAPTER 43

SRv6 Flex-Algo with TI-LFA and uLoop Avoidance

This document describes SRv6 Flex-Algo with TI-LFA and uLoop Avoidance.

- Feature Information for SRv6 Flex-Algo with TI-LFA and uLoop Avoidance, on page 455
- SRv6 Flex-Algo with TI-LFA and uLoop Avoidance, on page 455
- Prerequisites for SRv6 Flex-Algo with TI-LFA and uLoop Avoidance , on page 456
- Restrictions for SRv6 Flex-Algo with TI-LFA and uLoop Avoidance , on page 456
- Configure SRv6 IS-IS flexible algorithm, on page 457
- Traffic steering, on page 461

Feature Information for SRv6 Flex-Algo with TI-LFA and uLoop Avoidance

The table below provides the feature history for SRv6 Flex-Algo with TI-LFA and uLoop Avoidance.

Table 54:

Feature	Release	Description
SRv6 Flex-Algo with TI-LFA and uLoop Avoidance	Cisco IOS XE 17.18.1a	From Cisco IOS XE 17.18.1a, Flexible Algorithm enhances SRv6 by including functions like Topology Independent Loop-Free Alternate (TI-LFA) and microloop (uLoop) avoidance. This feature improves network resilience and efficiency.

SRv6 Flex-Algo with TI-LFA and uLoop Avoidance

A SRv6 flexible algorithm is a feature that

- extends the capabilities of SRv6.
- allows traffic engineering and optimization.

- improves network resilience and efficiency by incorporating functionalities like Topology Independent Loop-Free Alternate (TI-LFA) and uLoop (microloop) Avoidance.

In SRv6, the network forwards data using SRv6 SIDs and SRv6 Locators. SIDs are assigned from the locators, and forwarding entries for them exist only on the device where they are created. Locators help with network-wide reachability and forward traffic for the SIDs. Therefore, a locator represents an algorithm-specific path to a certain destination node in the SRv6 domain. Each node can advertise multiple SRv6 locators - one for each Flex-Algo it participates in, in addition to locators for the default algorithm (algo-0).

SRv6 flex-algo FRR and TI-LFA

Flex-Algo FRR and TI-LFA are network performance features that

- improve network performance by allowing network operators to compute backups within the Flex-Algo topology, and
- ensure that traffic can be rerouted quickly in the event of link or node failures.

The Forwarding Information Base (FIB) entries will include Flex-Algo specific End SIDs and End.X SIDs, which are essential for accurate traffic forwarding across the network.

SRv6 Flex-Algo Microloop-Avoidance

A Flex-Algo Microloop-Avoidance is a process that

- integrates specific processing into existing IPv6 uLoop routines,
- ensures that Forwarding Information Base (FIB) entries are tailored for each flexible algorithm by incorporating End SIDs and End.X SIDs, and
- is crucial for maintaining seamless traffic flow and avoiding microloops during network updates or changes.

Prerequisites for SRv6 Flex-Algo with TI-LFA and uLoop Avoidance

SRv6 must be enabled on the router before the Flexible Algorithm functionality is activated.

Restrictions for SRv6 Flex-Algo with TI-LFA and uLoop Avoidance

- Only up to 8 SRv6 flex-algo will be supported.
- SRv6 deployments are supported only at one level. Leaking locators between levels or redistribution of locators is not supported.

Configure SRv6 IS-IS flexible algorithm

TI-LFA and uLoop Avoidance settings are inherited from the Algo zero configuration in Flex-Algos. You cannot modify these settings for each Flex-Algo, instead, you can only disable them. For more information, see ISIS - SRv6: uLoop Avoidance and IPv6 Loop-Free Alternate Fast Reroute

Use the following configuration to set up SRv6 flex-algo. For more details, see, Enabling Segment Routing Flexible Algorithm

The following is an example of how to configure the SRv6 IS-IS flexible algorithm. The below example shows how to configure three flexible-algorithms. Algo 128 runs IGP metric, Algo 129 runs with delay metric and Algo runs with 133 TE metric.

```
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
ip address 1.1.1.1 255.255.255.255
ipv6 address 111::111/128
!
segment-routing srv6
locators
  locator loc1
   prefix FCCC:CCC1:A1::/48
   format usid-f3216
  locator F128
   prefix FCCC:F128:A1::/48
   format usid-f3216
   algorithm 128
  locator F129
   prefix FCCC:F129:A1::/48
   format usid-f3216
   algorithm 129
  locator F133
   prefix FCCC:F133:A1::/48
   format usid-f3216
   algorithm 133
!
router isis 1
net 50.1234.1721.1600.1001.00
is-type level-2-only
router-id Loopback0
metric-style wide
affinity-map BLACK bit-position 255
affinity-map BLUE bit-position 0
affinity-map GREEN bit-position 8
flex-algo 128
  advertise-definition
  priority 250
  affinity
   exclude-any
    name BLUE
    exit-fa-affinity-attr
   !
!
flex-algo 129
  advertise-definition
  metric-type delay
  priority 250
  affinity
   include-all
```

```
    name BLACK
!
flex-algo 133
  advertise-definition
  metric-type te
  priority 250
  affinity
   include-any
    name GREEN
!
address-family ipv6
  router-id Loopback0
  segment-routing srv6
   locator loc1
   locator F128
   locator F129
   locator F133
  fast-reroute per-prefix level-2 all
  fast-reroute ti-lfa level-2

interface Ethernet1/3
ipv6 address 15:15::1/64
ipv6 router isis 1
bfd interval 50 min_rx 50 multiplier 3
isis network point-to-point
isis te-metric flex-algo 10 level-2
isis ipv6 metric 10
isis ipv6 bfd
isis affinity flex-algo
  name BLACK
  name GREEN
  name BLUE

performance-measurement
interface Ethernet1/3
 delay-measurement
 next-hop ipv6 15:15::2
```

Verify SRv6 Flex-Algo with TI-LFA and uLoop Avoidance

Verify the configuration by using show commands to check the status and details of Flex-Algo settings:

The following is a sample output of the **show isis ipv6 rib flex-algo value** command showing all the IS-IS IPv6 local RIB information:

```
Router# sh isis ipv6 rib flex-algo 128

IS-IS IPv6 process 1, local RIB

Repair path attributes:
    DS - Downstream, LC - Linecard-Disjoint, NP - Node-Protecting
    PP - Primary-Path, SR - SRLG-Disjoint

Flex-algo 128

* CAFE:F128:606::/48
  Algorithm: 128
    via FE80::A8BB:CCFF:FE02:5E20/Ethernet0/2, type L2  metric 10 tag 0
     (installed)
     repair path: via FE80::A8BB:CCFF:FE02:5A10/Ethernet0/1 metric: 130 (DS,SR)
      TI-LFA link-protecting
        SRv6-Fwd-Id 25165861
        P node: r604 SID CAFE:0:604:: uN (PSP/USD)
```

```
        Q node: r605 SID CAFE:0:604:E002:: uA (PSP/USD)
      repair source: r606, metric to pfx: 150
```

The following is a sample output of the **show isis ipv6 flex-algo** command showing all the information regarding IS-IS IPv6 flexible algorithm running in multi-topology mode. If ISIS IPv6 routing runs in single-topology mode, use the **show isis flex-algo** command.

```
Router# show isis ipv6 flex-algo 128
Tag 1:
IS-IS Flex-Algo Database
 Flex-Algo count: 2
 Use delay metric advertisement: Application, Legacy
    IS-IS Level-2
      Delay metric: Inactive

Flex-Algo 128:
  Topology-2
    IS-IS Level-2
      Definition Priority: 128
      Definition Source: r601.00
      Definition Equal to Local: Yes
      Definition Metric Type: IGP
      Definition Flex-Algo Prefix Metric: No
      Disabled: No
      Microloop Avoidance Timer Running: No
   Local Priority: 128
   FRR Disabled: No
   Microloop Avoidance Disabled: No
```

The following is a sample output of the **show isis ipv6 topology flex-algovalue** command showing information regarding the IS-IS paths to intermediate systems running in multi-topology mode.If ISIS IPv6 routing runs in single-topology mode, use the command **show isis topology flex-algo**.

```
Router# sh isis ipv6 topology flex-algo 128

Tag 1:
Flex-algo 128
IS-IS TID 2 paths to level-2 routers
System Id            Metric     Next-Hop             Interface   SNPA
r601                 --
r602                 10         r602                 Et0/0       aabb.cc02.5a00
                                r602                 Et0/1       aabb.cc02.5a10
r603                 20         r602                 Et0/0       aabb.cc02.5a00
                                r602                 Et0/1       aabb.cc02.5a10
r604                 30         r602                 Et0/0       aabb.cc02.5a00
                                r602                 Et0/1       aabb.cc02.5a10
                                r606                 Et0/2       aabb.cc02.5e20
r605                 20         r606                 Et0/2       aabb.cc02.5e20
r606                 10         r606                 Et0/2       aabb.cc02.5e20
```

The following is a sample output of the **sh isis ipv6 fast-reroute summary flex-algo** command showing information regarding the IS-IS fast-reroute summary flex-algo:

```
Router# sh isis ipv6 fast-reroute summary flex-algo 128

Tag 1:
Flex-algo 128
IPv6 Fast-Reroute Protection Summary:

 Prefix Counts:          Total      Protected    Coverage
   High priority:        0          0            0%
   Normal priority:      18         12           66%
   Total:                18         12           66%
```

Use the **show isis ipv6 microloop-avoidance flex-algo** command to display the current uLoop flex-algo status.

```
Router# sh isis ipv6 microloop-avoidance flex-algo 128

Tag: 1
Algo  State            Delay  Running(L1/L2)
128   Segment-Routing  5000   NA/FALSE
```

Following is the example output for **show isis srv6 locators** command

```
Router# sh isis srv6 locators
ISIS SRv6 Locators:
Tag 1:

Name             Algo Prefix                                    Level
----             ---- ------                                    -----
A                0    CAFE:0:601::/48                           1-2
F128             128  CAFE:F128:601::/48                        1-2
```

Following is the example output for **show isis node locators** command

```
Router# sh isis node locators r601.00

Tag 1:

ISIS level-2 node information for r601.00
SRv6 locator cache:
 (MT-2) CAFE:0:601::/48, Algorithm: 0
    LSP 0, Index 1, Seq 0x45
    Metric: 0, Algoritm: 0
    End SID: CAFE:0:601:: uN (PSP/USD)
 (MT-2) CAFE:F128:601::/48, Algorithm: 128
    LSP 0, Index 1, Seq 0x45
    Metric: 0, Algoritm: 128
    End SID: CAFE:F128:601:: uN (PSP/USD)
```

Following is the example output for **sh isis node internal lsp locators** command

```
Router# sh isis node internal lsp locators r601.00

Tag 1:

ISIS level-2 node information for r601.00
 LSP r601.00-00 (1111.1111.1111.00-00) Seq 0x49 locator list:
 (MT-2) CAFE:F128:601::/48, Algo 128
 (MT-2) CAFE:0:601::/48, Algo 0
```

Monitor SRv6 Flex-Algo with TI-LFA and uLoop Avoidance

After configuring SRv6 Flex-Algo, monitor its performance and behavior using the following commands:

Use **show isis ipv6 fast-reroute summary flex-algo [number]** command to monitor Fast-reroute coverage and status.

Use **show isis ipv6 microloop-avoidance flex-algo [number]** command to check microloop avoidance settings.

Traffic steering

ISIS creates the Flex-Algo forwarding plane, which is different from the algo-0 forwarding plane due to metric and affinity constraints. The forwarding plane in SRv6 was established by installing locator IPv6 prefixes into the routing table. From Cisco IOS XE 17.17.1a, users can steer VPN traffic routed by BGP to Flex-Algo by instructing BGP to use a locator for the specific VRF that belongs to Flex-Algo. For example, locator F255 belongs to flex-algo 255 and is used for traffic in VRF V6.

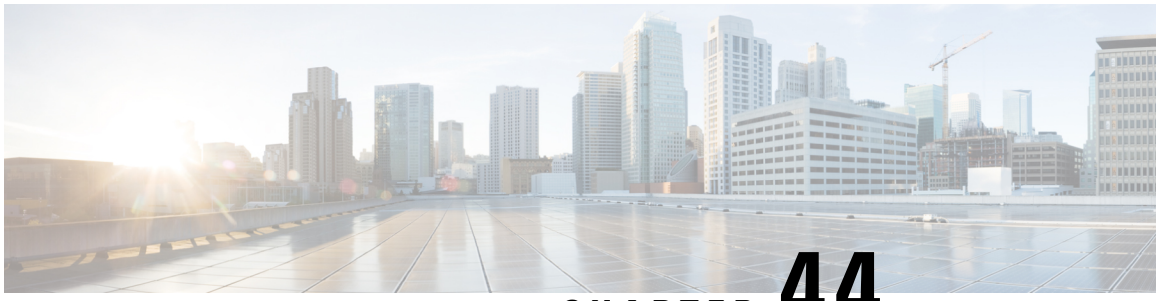
Configure traffic steering

The following sample configuration shows how to configure Traffic Steering

```
segment-routing srv6
locators
  locator F255
   prefix CAFE:F255:601::/48
   format usid-f3216
   algorithm 255

router bgp 100
..
address-family ipv6 vrf V6
  !
  segment-routing srv6
   locator F255
  exit-srv6
```

BGP than creates service SID which is used to steer traffic for BGP routes onto Flex-Algo forwarding plane.



CHAPTER 44

SRv6 Path MTU Discovery on IOS XE devices

This document describes SRv6 Path MTU Discovery on IOS XE devices.

- Feature Information SRv6 Path MTU Discovery , on page 463
- SRv6 Path MTU Discovery, on page 464
- Restrictions for SRv6 path MTU discovery, on page 466
- Configure SRv6 encapsulation MTU, on page 466
- Enable Transit-Node PTB relay, on page 466
- Verify SRv6 path MTU discovery, on page 467
- Supported SRv6 PMTU Commands, on page 468

Feature Information SRv6 Path MTU Discovery

The table below provides the feature history for SRv6 Path MTU Discovery.

Table 55:

Feature	Release	Description
SRv6 Path MTU Discovery	Cisco IOS XE 17.18.1a	This feature introduces a mechanism to determine the maximum transmission unit (MTU) for packets traversing an SRv6 underlay network. It ensures efficient packet forwarding by preventing fragmentation and packet drops, thereby allowing network devices to dynamically adjust packet sizes to avoid exceeding link MTU limits. The system relays ICMP Packet Too Big (PTB) messages from the SRv6 underlay to the IPv6/IPv4 overlay network, supporting both Transit-node and Headend-node PTB relay methods.

SRv6 Path MTU Discovery

SRv6 path MTU discovery is a mechanism that:

- addresses challenges in determining the maximum transmission unit (MTU) for packets traversing an SRv6 underlay network,
- ensures efficient packet forwarding by preventing fragmentation and packet drops, and
- helps network devices dynamically adjust packet sizes to avoid exceeding link MTU limits.

IPv6 fragmentation happens only at the source node, and intermediate nodes do not fragment packets. When an SRv6 ingress node encapsulates a packet that is too big, it drops the packet and sends an ICMPv6 Packet Too Big (PTB) message to the source. In the SRv6 L3VPN case, when the ICMPv6 PTB message reaches the head-end ingress PE, the PE node cannot forward the message to the actual source. This is because the PE cannot determine the correct VRF context from the ICMPv6 packet.

SRv6 PMTU discovery architecture

SRv6 PMTU discovery architecture is a design that:

- ensures compatibility with both SRv6 and non-SRv6 devices within the SRv6 core.
- supports interoperability across multiple vendors.
- enables relays ICMPv6 PTB messages through the network.

Relaying ICMP PTB messages

Relaying ICMP PTB messages is a process that:

- forwards ICMPv6 PTB error messages from the SRv6 underlay network to the original packet source,
- allows the source device to adjust its packet size to prevent further fragmentation or drops, and
- ensures proper communication of MTU limitations across different network segments.

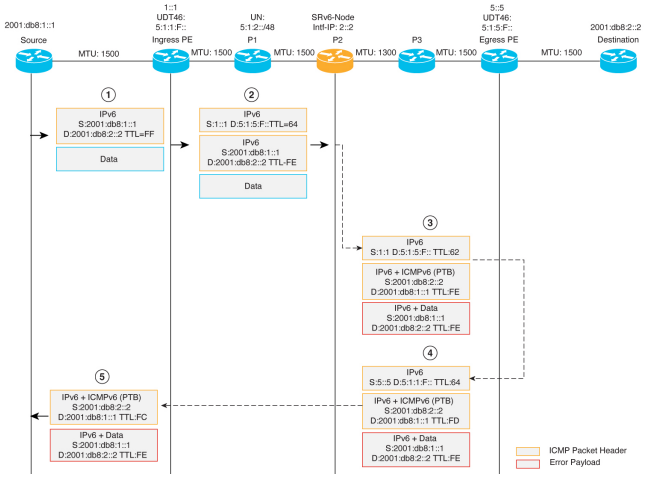
This process is essential because SRv6 does not assign a specific source address per VRF or per SRv6 policy, making it difficult for the ingress SRv6 device to identify the correct VRF to relay the ICMPv6 PTB message directly back to the source.

Transit-node PTB relay

Transit-node PTB relay is an implementation that:

- supports interoperability with IOS-XR devices,
- sends an ICMPv6 PTB message to the source device on the overlay network, and
- forwards the ICMP packet towards the egress PE using the received SRv6 outer header.

When a packet's MTU exceeds the link's MTU on an intermediate node, the intermediate node detects SRv6 encapsulation, calculates the PMTU value, generates the appropriate ICMP error message and then swaps the inner source and destination IP addresses before forwarding the packet towards the egress PE.

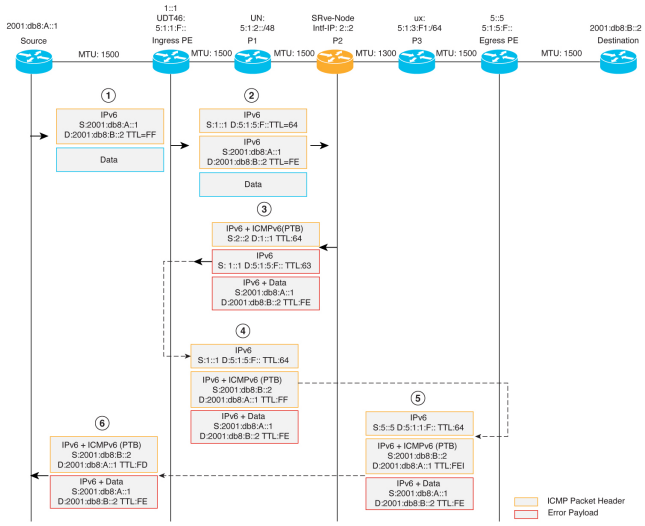


Headend-node PTB relay

Headend-node PTB relay is an implementation that

- supports intermediate devices that are non-SRv6 or SRv6 devices from other vendors,
- generates a new ICMPv6 PTB packet based on the inner header of the received ICMP PTB error message, and
- sends this new ICMP PTB message to the source device on the overlay network.

When a packet's MTU exceeds the link's MTU on an intermediate node, non-SRv6 nodes may send an ICMP PTB message towards the SRv6 source address (the SRv6 ingress PE). The SRv6 ingress PE then processes this message, generates a new ICMP PTB packet, and forwards it towards the egress PE using the SRv6 outer header from the received ICMP PTB message.



Restrictions for SRv6 path MTU discovery

The Transit-node PTB relay feature is supported only in a pure SRv6 deployment. It is not supported when the network includes a mix of IPv6-only nodes.

Configure SRv6 encapsulation MTU

Configuring SRv6 encapsulation MTU is a process that:

- involves setting the MTU value to include the SRv6 overhead,
- requires the MTU value to be based on your network requirements, and
- aims to prevent fragmentation and packet drops.

SRv6 encapsulation MTU command

Use the following command to configure the MTU:

```
segment-routing srv6
                   encapsulation
                   mtu <1280-9960>
```



Note If you prefer not to relay ICMP messages dynamically back to the source, use this command to send the ICMP message directly from the head-end. For IPv6 traffic and IPv4 traffic with the DF flag set, the head-end sends the ICMP message. For IPv4 traffic without the DF flag set, the head-end fragments the IPv4 packet before adding the SRv6 encapsulation.

Enable Transit-Node PTB relay

Enabling Transit-Node-PTB-Relay is a configuration process that:

- overrides the default Headend-Node-PTB-Relay behavior,
- activates the Transit-Node-PTB-Relay functionality, and
- serves interoperability purposes.

Headend-Node-PTB-Relay is the default behavior implemented.

Configuration command

Use the following configuration to enable Transit-Node-PTB-Relay:

```
segment-routing srv6
                   transit-node-ptb-relay
```

Verify SRv6 path MTU discovery

To provide commands and example outputs for verifying the platform's support, operational status, and counters of SRv6 Path MTU Discovery modes.

Verify platform support for SRv6 Path MTU Discovery modes

You can verify the platform's support for SRv6 Path MTU Discovery modes using the **show srv6 capabilities-parameters** command.

```
PE1#show srv6 capabilities-parameters
Platform Capabilities:
 SRv6:Yes
 PFP:Yes
 TILFA:Yes
 Transit-Node-PTB-Relay, Headend-Node-PTB-Relay
 Endpoint behaviors:
  uN (PSP/USD)
  uA (PSP/USD)
  uDT6
  uDT4
  uDT46
 Headend behaviors:
  H.ENCAP.RED
 Encap Parameters:
  Max-SL :16 (H-ENCAP-MSD:16 SL-MSD:16, END-POP-MSD:16 END-ENCAP-MSD:16)
  Encap :Collapsed
  Hop-limit propagate :Yes
  Traffic-class propagate :Yes
  Path-MTU supported :static
  SRH mode :
   combine :Yes
   stack :No
   insert :Yes

Parameters in-use:
 Encap Parameters:
  Source Address: C2:1::1, Loopback0 (Default)
  Hop-Limit: 64 (Default)
  Traffic-class: 0 (Default)
  MTU:  0 (Default IPv4 MTU)
```

Verify SRv6 Path MTU counters for Transit-Node PTB Relay

The following output shows SRv6 Path MTU counters for Transit-Node PTB Relay.

```
P4#show srv6 pmtu counters
 PMTU Transit-Relay is                                Enabled
 PMTU Headend-Remote-Relay is              Enabled
 PMTU Headend-Local-Relay is                  Enabled
 PMTU SRv6 not enabled:                          16
 PMTU Handle ICMP6 PTB Calls:              11
 Matched Block for SRv6 Encap:                11
 ICMP6 Transit PTB Gen:                           1
 ICMP4 Transit PTB Gen:                           3
 IPv6 MTU change needed:                        1
 IPv4 MTU change needed:                       10
 ICMP4 Internal ICMP Build Error:               2
 IPv4 DF bit is not set:                                5
 PMTU Transit-Relay disabled hit:             15
```

```
 IPv6 Last MTU: 1360 Src: 1700::7 Dst: 6800::8 Originator 1411::4 Transit [06-16 00:52:01.236]

 IPv6 Min  MTU: 1360 Src: 1700::7 Dst: 6800::8 Originator 1411::4 Transit [06-16 00:52:01.236]

 IPv4 Last MTU: 1360 Src: 17.0.0.7 Dst: 68.0.0.8 Originator 1411::4 Transit DF  [06-16
00:51:45.966]
 IPv4 Min  MTU: 1360 Src: 17.0.0.7 Dst: 68.0.0.8 Originator 1411::4 Transit [06-16
00:51:09.587]
 SRv6 FIB Queue Size:             100
 SRv6 FIB Queue Peak:             1
 SRv6 FIB Queue Drops:            0
 SRv6 FIB Queue Enqueues:         11
```

Verify SRv6 Path MTU counters for Headend-Node PTB Relay

The following output shows SRv6 Path MTU counters for Headend-Node PTB Relay.

```
PE1#show srv6 pmtu counters
 PMTU Transit-Relay is                                Disabled
 PMTU Headend-Remote-Relay is               Enabled
 PMTU Headend-Local-Relay is                   Enabled
 PMTU Handle ICMP6 PTB Calls:               11
 Found SRH for SRv6 Encap:                      11
 ICMP6 Headend-Local PTB Gen:               1
 ICMP4 Headend-Local PTB Gen:               3
 IPv6 MTU change needed:                         1
 IPv4 MTU change needed:                         10
 ICMP4 Internal ICMP Build Error:                2
 IPv4 DF bit is not set:                                  5
 PMTU counters cleared:                                4
 IPv6 Last MTU: 1336 Src: 1700::7 Dst: 6800::8 Originator A001::1 Headend-Local VRF: vpn-101
 [06-15 14:26:31.229]
 IPv6 Min  MTU: 1336 Src: 1700::7 Dst: 6800::8 Originator A001::1 Headend-Local VRF: vpn-101
 [06-15 14:26:31.229]
 IPv4 Last MTU: 1336 Src: 17.0.0.7 Dst: 68.0.0.8 Originator A001::1 Headend-Local VRF:
vpn-101 DF  [06-15 14:26:21.635]
 IPv4 Min  MTU: 1336 Src: 17.0.0.7 Dst: 68.0.0.8 Originator A001::1 Headend-Local VRF:
vpn-101 [06-15 14:26:03.760]
 SRv6 FIB Queue Size:             100
 SRv6 FIB Queue Peak:             1
 SRv6 FIB Queue Drops:            0
 SRv6 FIB Queue Enqueues:         11
```

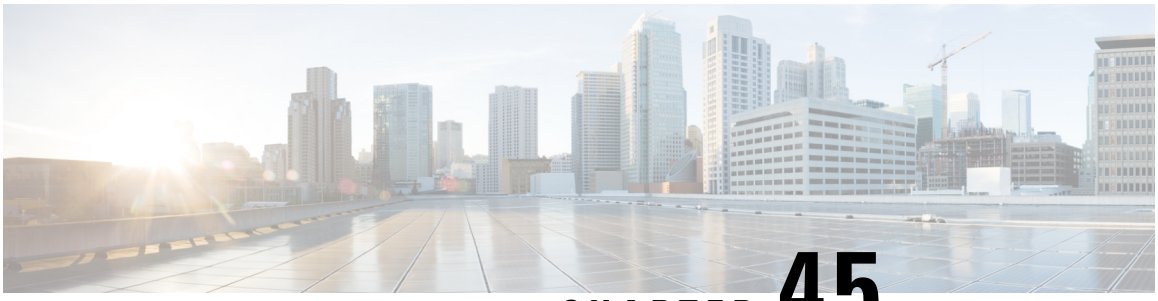
Supported SRv6 PMTU Commands

You can manage and troubleshoot SRv6 PMTU features using the following commands:

Table 56: SRv6 PMTU Commands

Command	Description
clear segment-routing counters pmtu	Clears SRv6 PMTU statistics.
debug srv6 pmtu	Enables debugging for SRv6 PMTU.
show srv6 capabilities-parameters	Verifies a platform's support for SRv6 Path MTU Discovery.

Command	Description
show srv6 pmtu counters	Displays SRv6 Path MTU counters.



CHAPTER 45

MVPN ingress replication over SRv6

This document describes MVPN Ingress Replication (IR) over SRv6.

- Feature Information for MVPN ingress replication over SRv6, on page 471
- MVPN Ingress Replication over SRv6, on page 471
- How MVPN Ingress Replication over SRv6 works, on page 472
- Partitioned MDT profiles, on page 473
- Configure MVPN ingress replication over SRv6 , on page 473
- Configure SRv6 MVPN multicast, on page 474
- Verify and monitor MVPN ingress replication over SRv6, on page 476
- Appendix: configuration reference, on page 487

Feature Information for MVPN ingress replication over SRv6

The table below provides the feature history for MVPN ingress replication over SRv6.

Table 57:

Feature	Release	Description
MVPN Ingress Replication (IR) over SRv6	Cisco IOS XE 17.18.1a	This feature enables the transport of IPv4 Multicast traffic across an SRv6 network. It simplifies multicast deployment by using the existing SRv6 unicast infrastructure as the underlay. With this feature, the ingress PE router receives multicast traffic and creates a separate unicast SRv6-encapsulated copy for each egress PE router in the multicast group.

MVPN Ingress Replication over SRv6

This document describes MVPN Ingress Replication (IR) over SRv6, a feature that

- enables the transport of IPv4 Multicast traffic across an SRv6 network,
- leverages the existing SRv6 unicast infrastructure
- simplifies multicast deployment by using SRv6 as the underlay transport.

The ingress PE router receives multicast traffic and creates a separate unicast SRv6-encapsulated copy for each egress PE router in the multicast group.

- Each packet is encapsulated using an SRv6 encapsulation that guides it to the intended egress PE.
- The core network simply forwards these SRv6 packets based on their segment routing headers; no multicast state is maintained in the core.
- The egress PE receives and decapsulates the outer SRv6 encapsulation, and then forwards the multicast data in corresponding VPN.

How MVPN Ingress Replication over SRv6 works

This feature primarily supports IPv4 MVPN over an SRv6 network, specifically implementing the SRv6 End.DTMC4 behavior. Key capabilities and benefits include:

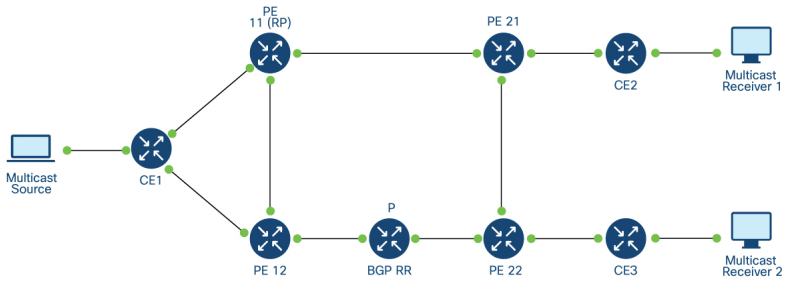
Summary

The key components involved in the MVPN Ingress Replication over SRv6 process are:

- Ingress CE: The customer edge device that faces the multicast source.
- Ingress PE: The provider edge router that receives multicast traffic, performs replication, and encapsulates it.
- Core Network (P routers): The underlying SRv6 fabric that forwards encapsulated packets without maintaining multicast state.
- Egress PE: The provider edge router that receives and decapsulates multicast traffic for receivers.
- Egress CE: The customer edge device that faces the multicast receiver.

MVPN Ingress Replication over SRv6 describes the end-to-end flow of IPv4 multicast traffic from source to receiver using SRv6 as the underlay, leveraging ingress replication at the PE routers.

Workflow



The MVPN Ingress Replication over SRv6 process involves the following stages:

1. The ingress CE faces the multicast source and forwards the multicast traffic to the Ingress PE. The Ingress PE performs replication based on learned multicast routes and encapsulates each replica in an IPv6 header. The Egress PE allocates a SID for MVPN per VRF, and the Ingress PE uses that SID for all multicast routes in the VRF for forwarding in the underlay. Each IPv6 packet is then forwarded to the appropriate Egress PE.
2. The Egress PE receives the unicast IPv6 packet destined for its locally allocated SRv6 SID, decapsulates the outer SRv6 header, and performs a multicast lookup in the associated VRF. The payload is then delivered to the Egress CE facing the multicast receiver.
3. The Provider Edge (PE) and Provider (P) routers form the SRv6 fabric, while PE devices act as SRv6 edge nodes.

Partitioned MDT profiles

In a partitioned MDT profile, only those egress PE routers with interested receivers are associated with that ingress PE. By using Partitioned MDT we limit the replication state in the Ingress PE to only include Egress PEs that contain receivers.

When a Partitioned MDT profile is used in PE-PE IR, egress PEs may need to identify traffic from ingress PEs, which uses an internal RPF-ID.

Configure MVPN ingress replication over SRv6

Enable MVPN Ingress Replication over SRv6 for the specified VRF, allowing multicast traffic transport over the SRv6 underlay.

SRv6 MVPN IR transports only IPv4 multicast traffic over an SRv6 network. Intra-as BGP auto-discovery for SRv6 and C-mroute overlay signalling are enabled by default. Strict RPF check is also enabled by default.

Before you begin

For more information on static RP and Auto RP configuration, refer Configuring Basic IP Multicast

Follow these steps to configure MVPN IR over SRv6.

Procedure

Step 1 Access the VRF definition configuration mode.

Example:

```
vrf definition BLUE
```

Step 2 Enter the address-family configuration mode for IPv4.

Example:

```
address-family ipv4
```

Step 3 Configure SRv6 multicast ingress replication.

Example:

```
srv6-mcast ingress-replication partitioned [data]
```

- `partitioned`: Specifies partitioned MDT.
- `data`: Specifies data MDT.

MVPN Ingress Replication over SRv6 is configured for the specified VRF, enabling multicast traffic transport over the SRv6 underlay.

Configure SRv6 MVPN multicast

To enable MVPN Ingress Replication over SRv6 for multicast traffic within a VRF.

Intra-as BGP auto-discovery for SRv6 and C-mroute overlay signaling are enabled by default. Strict RPF check is also enabled by default.

Before you begin

Follow these steps to enable MVPN Ingress Replication over SRv6 for multicast traffic within a VRF.

Procedure

Step 1 Enter the global configuration mode and access the VRF definition:

Example:

```
vrf definition <vrf-name>
```

Step 2 Configure the Route Distinguisher (RD) for the VRF.

Example:

```
rd <route-distinguisher>
```

Step 3 For IPv4 multicast, enter the address-family.

Example:

```
address-family ipv4
```

Step 4 Configure the route-target for the VRF's address-family.

Example:

```
route-target {import | export | both } <route-target-ext-community>
```

Step 5 Enable SRv6 multicast ingress replication:

Example:

```
srv6-mcast ingress-replication partitioned [data]
```

Step 6 Exit the IPv4 address-family configuration mode:

Example:

```
exit-address-family
```

The specified VRF is configured to use SRv6 for MVPN Ingress Replication.

Configure BGP MVPNv4 neighbor encapsulation

To specify the encapsulation type for MVPNv4 updates advertised to a BGP neighbor.

Before you begin

Follow these steps to specify the encapsulation type for MVPNv4 updates advertised to a BGP neighbor.

Procedure

Step 1 Enter the global configuration mode and access the BGP router configuration.

Example:

```
router bgp <autonomous-system-number>
```

Step 2 Enter the IPv4 MVPN address-family configuration mode.

Example:

```
address-family ipv4 mvpn
```

Step 3 Configure the encapsulation option for a specific neighbor.

Example:

```
address-family ipv4 mvpn
            neighbor <neighbor-address> activate
            neighbor <neighbor-address> send-community both
```

The BGP neighbor is configured to advertise MVPNv4 updates only for the specified encapsulation type.

BGP SRv6 configuration

To illustrate a sample BGP configuration for SRv6 in a multicast environment.

This example displays a BGP configuration that

- includes SRv6 locator and
- allocation mode settings, which are utilized by Multicast.

```
router bgp <autonomous-system-number>
          neighbor <neighbor-address> remote-as <remote-as-number>
          neighbor <neighbor-address> update-source <interface-type-and-number>
          segment-routing srv6
          locator <locator-name>
          exit-srv6
          address-family vpnv4
          neighbor <neighbor-address> activate
          neighbor <neighbor-address> send-community both
          segment-routing srv6
```

```
locator <locator-name>
alloc-mode per-vrf
exit-srv6
address-family ipv4 vrf <vrf-name>
segment-routing srv6
locator <locator-name>
alloc-mode per-vrf
exit-srv6
address-family ipv4 mvpn
neighbor <neighbor-address> activate
neighbor <neighbor-address> send-community both
```

Verify and monitor MVPN ingress replication over SRv6

To provide commands and examples for monitoring and verifying the operation of SRv6 MVPN.

These commands help you monitor and verify the operation of SRv6 MVPN.

-

The following example shows the output of the **show ip mroute** command on an Ingress PE router to display the multicast routing table for a specific VRF and group address.

```
PE12#show ip mroute vrf vpn-11 239.1.1.2 verbose
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf,
       e - encap-helper tunnel flag, l - LISP decap ref count contributor
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
                          t - LISP transit group
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.1.1.2), 00:04:47/stopped, RP 11.11.11.11, flags: SP
  Incoming interface: Srvif1, RPF nbr A002::1
  Remote RD: 11:11  RPF-ID: 0xA002::1
  Outgoing interface list: Null

(11.1.0.2, 239.1.1.2), 00:04:47/00:02:07, flags: TyGq
  Incoming interface: GigabitEthernet2.11, RPF nbr 14.1.0.1
   MDT TX nr: 2097162  LSM-ID: 0x5E
  Outgoing interface list:
    Srvif1, LSM MDT: 5E (data), Forward/Sparse, 00:04:47/stopped, Pkts:0, flags:
```

The following output example displays multicast routing entry details and MVPN replication.

```
PE12#show ip mroute vrf vpn-11 239.1.1.2 11.1.0.2 verbose

(11.1.0.2, 239.1.1.2), 00:02:32/00:00:27, flags: TyGq
  Incoming interface: GigabitEthernet2.11, RPF nbr 14.1.0.1
```

```
   MDT TX nr: 2097162  LSM-ID: 0x5E
  Outgoing interface list:
    Srvif1, LSM MDT: 5E (data), Forward/Sparse, 00:02:32/00:00:27, Pkts:0, flags:
```

The following example shows the output of the **show ip mfib** command on an Ingress PE router, displaying the multicast forwarding information base (MFIB) for a specific VRF and group address.

```
PE12#show ip mfib vrf vpn-11 239.1.1.2
Entry Flags:    C - Directly Connected, S - Signal, IA - Inherit A flag,
                ET - Data Rate Exceeds Threshold, K - Keepalive
                DDE - Data Driven Event, HW - Hardware Installed
                ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
                MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
                MS  - MoFRR  Entry in Sync, MC - MoFRR entry in MoFRR Client,
                e   - Encap helper tunnel flag.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
                MA - MFIB Accept, A2 - Accept backup,
                RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   HW Pkt Count/FS Pkt Count/PS Pkt Count   Egress Rate in pps
VRF vpn-11
 (*,239.1.1.2) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   0/0/0/0, Other: 0/0/0
   Srvif1, SRV6 LSM/5D, RPF-ID: 0xA002::1, Flags: A
 (11.1.0.2,239.1.1.2) Flags: ET HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   607/2/200/3, Other: 27/27/0
   GigabitEthernet2.11 Flags: A
   Srvif1, SRV6 LSM/5E, RPF-ID: *, Flags: F
     Pkts: 0/0/0    Rate: 0 pps
```

The **show bgp ipv4 mvpn vrf vpn-11 detail** command displays detailed BGP MVPN routing table entries for VRF vpn-11.

```
PE12#show bgp ipv4 mvpn vrf vpn-11 detail

Route Distinguisher: 11:11 (default for vrf vpn-11)
BGP routing table entry for [1][11:11][A002::1]/24, version 3764603
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A002::1 (metric 20) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 2.2.2.2, Cluster list: 5.5.5.5
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [1][11:11][A003::1]/24, version 3764604
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A003::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 3.3.3.3, Cluster list: 5.5.5.5
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 4 2025 05:52:41 UTC
BGP routing table entry for [1][11:11][A004::1]/24, version 3764605
```

```
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     23
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (4.4.4.4)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:11:11
      rx pathid: 0, tx pathid: 0x0
      Updated on Jul 30 2025 04:45:12 UTC
BGP routing table entry for [1][11:11][A006::1]/24, version 3764606
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A006::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 6.6.6.6, Cluster list: 5.5.5.5
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 4 2025 05:52:41 UTC
BGP routing table entry for [3][11:11][*][*][A002::1]/26, version 3764768
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A002::1 (metric 20) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 2.2.2.2, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A002::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [3][11:11][*][*][A003::1]/26, version 3764769
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A003::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 3.3.3.3, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A003::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 4 2025 05:52:41 UTC
BGP routing table entry for [3][11:11][*][*][A004::1]/26, version 3764770
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     23
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (4.4.4.4)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:11:11
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A004::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Jul 30 2025 04:45:12 UTC
BGP routing table entry for [3][11:11][*][*][A006::1]/26, version 3764771
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
```

```
  Local
    A006::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 6.6.6.6, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A006::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 4 2025 05:52:41 UTC
BGP routing table entry for [3][11:11][*][224.0.1.39][A002::1]/30, version 3764772
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A002::1 (metric 20) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 2.2.2.2, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A002::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [3][11:11][*][224.0.1.40][A002::1]/30, version 3764773
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A002::1 (metric 20) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 2.2.2.2, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A002::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [3][11:11][11.1.0.2][239.1.1.2][A004::1]/34, version 3764774
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     23
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (4.4.4.4)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:11:11
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A004::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 09:22:19 UTC
BGP routing table entry for [4]{[3][11:11][*][*][A002::1]/26}[A004::1]/44, version 3739440
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     23
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (4.4.4.4)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      IPv6-Extended Community: RT::A002::1:0
      PMSI Attribute: Flags: 0x0, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A004::1 >
      srv6 in-sid: F:F128:4:E000::
      rx pathid: 3, tx pathid: 0x0
      Updated on Aug 8 2025 01:30:45 UTC
BGP routing table entry for [4]{[3][11:11][*][224.0.1.39][A002::1]/30}[A004::1]/48, version
 3666900
```

```
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     23
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (4.4.4.4)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      IPv6-Extended Community: RT::A002::1:0
      PMSI Attribute: Flags: 0x0, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A004::1 >
      srv6 in-sid: F:F128:4:E000::
      rx pathid: 3, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [4]{[3][11:11][*][224.0.1.40][A002::1]/30}[A004::1]/48, version
 3666901
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     23
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (4.4.4.4)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      IPv6-Extended Community: RT::A002::1:0
      PMSI Attribute: Flags: 0x0, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A004::1 >
      srv6 in-sid: F:F128:4:E000::
      rx pathid: 3, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [5][11:11][11.1.0.2][239.1.1.2]/18, version 3598737
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     23
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (4.4.4.4)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:11:11
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 09:22:12 UTC
BGP routing table entry for [7][11:11][100][11.1.0.2/32][239.1.1.2/32]/22, version 3730109
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A006::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:4.4.4.4:3
      Originator: 6.6.6.6, Cluster list: 5.5.5.5
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 8 2025 00:54:16 UTC
```

Use the **show mvpn ipv4 vrf vpn-11 auto-discovery detail** command to display a summary of the MVPN auto-discovery routes.

```
PE12#show mvpn ipv4 vrf vpn-11 auto-discovery detail
I-PMSI - Intra-AS Inclusive-PMSI, S-PMSI - Selective-PMSI
* - Indicates Wildcard source or group address

 [I-PMSI][11:11][A002::1], Joined
  Orig: Remote Uptime: 2w5d Type: NONE

 [I-PMSI][11:11][A003::1], Joined
  Orig: Remote Uptime: 18:31:19 Type: NONE
```

```
[I-PMSI][11:11][A006::1], Joined
 Orig: Remote Uptime: 18:31:19 Type: NONE

[S-PMSI][11:11][*][*][A002::1], Joined
 Orig: Remote Uptime: 6d00h Type: IR SRV6

[S-PMSI][11:11][*][*][A003::1],
 Orig: Remote Uptime: 18:31:19 Type: IR SRV6

[S-PMSI][11:11][*][*][A006::1],
 Orig: Remote Uptime: 18:31:19 Type: IR SRV6

[S-PMSI][11:11][*][224.0.1.39][A002::1], Joined
 Orig: Remote Uptime: 6d00h Type: IR SRV6

[S-PMSI][11:11][*][224.0.1.40][A002::1], Joined
 Orig: Remote Uptime: 6d00h Type: IR SRV6

[I-PMSI][11:11][A004::1], Joined
 Orig: Local Uptime: 5d19h Type: NONE

[S-PMSI][11:11][*][*][A004::1], Joined
 Orig: Local Uptime: 5d19h Type: IR SRV6

[S-PMSI][11:11][11.1.0.2][239.1.1.2][A004::1], Joined
 Orig: Local Uptime: 00:01:04 Type: IR SRV6
```

Use the **show mvpn ipv4 vrf vpn-11 leaf-information detail** command to display a summary of the MVPN leaf-information details.

```
PE12#show mvpn ipv4 vrf vpn-11 leaf-information detail
Route-Type: 1 - Intra-AS Inclusive-PMSI, 3 - Selective-PMSI
* - Indicates Wildcard source or group address

 [3][11:11][*][*][A004::1], Joined
  Remote Orig: A003::1 UMH: 0.0.0.0 Uptime: 00:01:41 Type: IR SRV6
  Remote SID:{F:F128:3:E024::, pl:0 bl:32 nl:16 fl:16 al:0 st:0 bhv:76} IR-address: A003::1
 Replication-Id: 200000

 [3][11:11][*][*][A004::1], Joined
  Remote Orig: A006::1 UMH: 0.0.0.0 Uptime: 00:01:24 Type: IR SRV6
  Remote SID:{F:F128:6:E024::, pl:0 bl:32 nl:16 fl:16 al:0 st:0 bhv:76} IR-address: A006::1
 Replication-Id: 200000

 [3][11:11][11.1.0.2][239.1.1.2][A004::1], Joined
  Remote Orig: A003::1 UMH: 0.0.0.0 Uptime: 00:01:33 Type: IR SRV6
  Remote SID:{F:F128:3:E024::, pl:0 bl:32 nl:16 fl:16 al:0 st:0 bhv:76} IR-address: A003::1
 Replication-Id: 20000A

 [3][11:11][11.1.0.2][239.1.1.2][A004::1], Joined
  Remote Orig: A006::1 UMH: 0.0.0.0 Uptime: 00:01:24 Type: IR SRV6
  Remote SID:{F:F128:6:E024::, pl:0 bl:32 nl:16 fl:16 al:0 st:0 bhv:76} IR-address: A006::1
 Replication-Id: 20000A

 [3][11:11][*][*][A002::1], Joined
  Local Orig: A004::1 UMH: A002::1 Uptime: 00:01:41 Type: IR SRV6
  Local SID:{F:F128:4:E000::, pl:64 bl:32 nl:16 fl:16 al:0 st:0 bhv:76} IR-address: A004::1
 IR-Egress-PSM-Id: 41947810

 [3][11:11][*][224.0.1.39][A002::1], Joined
  Local Orig: A004::1 UMH: A002::1 Uptime: 5d19h Type: IR SRV6
  Local SID:{F:F128:4:E000::, pl:64 bl:32 nl:16 fl:16 al:0 st:0 bhv:76} IR-address: A004::1
 IR-Egress-PSM-Id: 41947810
```

```
 [3][11:11][*][224.0.1.40][A002::1], Joined
  Local Orig: A004::1 UMH: A002::1 Uptime: 5d19h Type: IR SRV6
  Local SID:{F:F128:4:E000::, pl:64 bl:32 nl:16 fl:16 al:0 st:0 bhv:76} IR-address: A004::1
 IR-Egress-PSM-Id: 41947810
```

To view information on the replication branches, including remote SIDs, for the specified multicast LSM ID use the command below.



Note The LSM MDT: <value> field from the above **show ip mroute** output, for instance 5E in this example, provides the LSM ID value of the below **show mvpn replication** command.

```
PE12#show mvpn replication lsm-id 5E

Repl ID : 20000A   LSM ID : 5E   Uptime : 00:03:03
  Path Set ID         : 2800FD6
  Replication branches: 2
   IR (A003::1)
      Uptime         : 00:03:03      Refcount  : 1
      Remote SID: F:F128:3:E024::
   IR (A006::1)
      Uptime         : 00:02:54      Refcount  : 1
      Remote SID: F:F128:6:E024::
```

Use the following command to display detailed BGP MVPN information for the specified VRF.

```
PE22#show bgp ipv4 mvpn vrf vpn-11 detail

Route Distinguisher: 11:11 (default for vrf vpn-11)
BGP routing table entry for [1][11:11][A002::1]/24, version 1865054
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A002::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 2.2.2.2, Cluster list: 5.5.5.5
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [1][11:11][A003::1]/24, version 1865055
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A003::1 (metric 20) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 3.3.3.3, Cluster list: 5.5.5.5
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 4 2025 05:52:41 UTC
BGP routing table entry for [1][11:11][A004::1]/24, version 1865056
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A004::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 4.4.4.4, Cluster list: 5.5.5.5
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
```

```
BGP routing table entry for [1][11:11][A006::1]/24, version 1865057
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     9
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (6.6.6.6)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:11:11
      rx pathid: 0, tx pathid: 0x0
      Updated on Jul 16 2025 15:42:02 UTC
BGP routing table entry for [3][11:11][*][*][A002::1]/26, version 1865219
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A002::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 2.2.2.2, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A002::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [3][11:11][*][*][A003::1]/26, version 1865220
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A003::1 (metric 20) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 3.3.3.3, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A003::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 4 2025 05:52:41 UTC
BGP routing table entry for [3][11:11][*][*][A004::1]/26, version 1865221
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A004::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 4.4.4.4, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A004::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [3][11:11][*][*][A006::1]/26, version 1865222
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     9
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (6.6.6.6)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:11:11
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A006::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Jul 16 2025 15:42:02 UTC
BGP routing table entry for [3][11:11][*][224.0.1.39][A002::1]/30, version 1865223
```

```
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A002::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 2.2.2.2, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A002::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [3][11:11][*][224.0.1.40][A002::1]/30, version 1865224
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A002::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 2.2.2.2, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A002::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [3][11:11][11.1.0.2][239.1.1.2][A004::1]/34, version 1865225
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A004::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 4.4.4.4, Cluster list: 5.5.5.5
      PMSI Attribute: Flags: 0x1, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A004::1 >
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [4]{[3][11:11][*][*][A002::1]/26}[A006::1]/44, version 1848708
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     9
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (6.6.6.6)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      IPv6-Extended Community: RT::A002::1:0
      PMSI Attribute: Flags: 0x0, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A006::1 >
      srv6 in-sid: F:F128:6:E024::
      rx pathid: 3, tx pathid: 0x0
      Updated on Aug 8 2025 00:54:16 UTC
BGP routing table entry for [4]{[3][11:11][*][*][A004::1]/26}[A006::1]/44, version 1848705
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     9
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (6.6.6.6)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      IPv6-Extended Community: RT::A004::1:0
      PMSI Attribute: Flags: 0x0, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A006::1 >
      srv6 in-sid: F:F128:6:E024::
```

```
      rx pathid: 3, tx pathid: 0x0
      Updated on Aug 8 2025 00:54:16 UTC
BGP routing table entry for [4]{[3][11:11][*][224.0.1.39][A002::1]/30}[A006::1]/48, version
 1822559
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     9
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (6.6.6.6)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      IPv6-Extended Community: RT::A002::1:0
      PMSI Attribute: Flags: 0x0, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A006::1 >
      srv6 in-sid: F:F128:6:E024::
      rx pathid: 3, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [4]{[3][11:11][*][224.0.1.40][A002::1]/30}[A006::1]/48, version
 1822560
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     9
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (6.6.6.6)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      IPv6-Extended Community: RT::A002::1:0
      PMSI Attribute: Flags: 0x0, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A006::1 >
      srv6 in-sid: F:F128:6:E024::
      rx pathid: 3, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [4]{[3][11:11][11.1.0.2][239.1.1.2][A004::1]/34}[A006::1]/52,
version 1848706
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     9
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (6.6.6.6)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      IPv6-Extended Community: RT::A004::1:0
      PMSI Attribute: Flags: 0x0, Tunnel type: IR, length 16, label:exp-null tunnel
identifier: < Tunnel Endpoint: A006::1 >
      srv6 in-sid: F:F128:6:E024::
      rx pathid: 3, tx pathid: 0x0
      Updated on Aug 8 2025 00:54:16 UTC
BGP routing table entry for [5][11:11][11.1.0.2][239.1.1.2]/18, version 1822081
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Flag: 0x100
  Not advertised to any peer
  Refresh Epoch 2
  Local
    A004::1 (metric 30) from A005::1 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:11:11
      Originator: 4.4.4.4, Cluster list: 5.5.5.5
      rx pathid: 0, tx pathid: 0x0
      Updated on Aug 7 2025 17:53:45 UTC
BGP routing table entry for [6][11:11][100][11.11.11.11/32][239.1.1.2/32]/22, version 1848709

  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     9
```

```
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (6.6.6.6)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:2.2.2.2:3
      rx pathid: 3, tx pathid: 0x0
      Updated on Aug 8 2025 00:54:16 UTC
BGP routing table entry for [7][11:11][100][11.1.0.2/32][239.1.1.2/32]/22, version 1848707
  Paths: (1 available, best #1, table MVPNv4-BGP-Table)
  Advertised to update-groups:
     9
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (6.6.6.6)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:4.4.4.4:3
      rx pathid: 3, tx pathid: 0x0
      Updated on Aug 8 2025 00:54:16 UTC
```

The following example shows the output of the **show ip rpf** command on an Egress PE router, displaying the RPF ID (strict RPF check) of multicast routing table for a specific VRF and multicast group address.

```
PE22#show ip rpf vrf vpn-11 11.1.0.2
RPF information for ? (11.1.0.2)
  RPF interface: Srvif1
  RPF neighbor: ? (A004::1)
  RPF route/mask: 11.1.0.0/16
  RPF type: unicast (bgp 100)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv6 unicast base
```

The following example shows the output of the **show ip mfib** command on an Egress PE router, displaying multicast forwarding information for a specific VRF and multicast group address.

```
PE22#show ip mfib vrf vpn-11 239.1.1.2
Entry Flags:    C - Directly Connected, S - Signal, IA - Inherit A flag,
                ET - Data Rate Exceeds Threshold, K - Keepalive
                DDE - Data Driven Event, HW - Hardware Installed
                ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
                MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
                MS  - MoFRR  Entry in Sync, MC - MoFRR entry in MoFRR Client,
                e   - Encap helper tunnel flag.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
                MA - MFIB Accept, A2 - Accept backup,
                RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   HW Pkt Count/FS Pkt Count/PS Pkt Count   Egress Rate in pps
VRF vpn-11
 (*,239.1.1.2) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   0/0/0/0, Other: 0/0/0
   Srvif1, SRV6 LSM/0, RPF-ID: 0xA002::1, Flags: A
   GigabitEthernet3.11 Flags: F NS
     Pkts: 0/0/0    Rate: 0 pps
 (11.1.0.2,239.1.1.2) Flags: HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   654/2/200/3, Other: 0/0/0
   Srvif1, SRV6 LSM/0, RPF-ID: 0xA004::1, Flags: A
   GigabitEthernet3.11 Flags: F NS
     Pkts: 0/0/0    Rate: 0 pps
```

The following example output displays detailed SID multicast information.

```
PE22# show segment-routing srv6 sid F:F128:6:E024:: detail

SID: F:F128:6:E024::     Type: DYNAMIC
 Behavior: End.DTMC4 (76)
 Context:
    interface: (not-set)
    vrf: (not-set), v4-topo-id: 0xFFFF, v6-topo-id: 0xFFFF
    next-hop: (not-set)
    policy: (not-set)
    distinguisher: (not-set)
    protected: n/a
    MCAST Instance: 1
      vrf v4: vpn-11, topo-id: 0x3
      ps_fwd_id: 41943073
 Stats:
    Packets: 12392565   Bytes: 2471022400
 User:
    User:Refcount                       Locator:Refcount
    -------------                       ----------------
    bgp-mvpn(2):1                       F128:1
 User Event history:
    Timestamp                  Client                         Event type
    ---------                  ------                         ----------
    07-16 15:41:51.677         bgp-mvpn(2)                    ALLOC
```

Appendix: configuration reference

SRv6 configuration for PE11

This section details the configuration of PE11, a Provider Edge router, focusing on its role in an SRv6 enabled network with advanced traffic engineering and VPN services

```
hostname PE11

vrf definition vpn-11
 rd 11:11
 route-target export 11:11
 route-target import 11:11
 !
 address-family ipv4
  srv6-mcast ingress-replication partitioned data
  route-target export 11:11
  route-target import 11:11
 exit-address-family
 !
 address-family ipv6
  route-target export 11:11
  route-target import 11:11
 exit-address-family
!
ip multicast-routing vrf vpn-11 distributed

bfd-template single-hop bfd_tmp
 interval microseconds min-tx 7000 min-rx 7000 multiplier 3
!
!
!
```

```
!
!
!
crypto ikev2 keyring gre_key
 peer gre
  address ::/0
  pre-shared-key cisco123
 !
!
!
crypto ikev2 profile gre_profile
 match identity remote address ::/0
 authentication remote pre-share
 authentication local pre-share
 keyring local gre_key
!
!
crypto ipsec transform-set gre_transet esp-aes esp-sha-hmac
 mode transport
!
!
crypto ipsec profile gre_profile
 set transform-set gre_transet
 set ikev2-profile gre_profile
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
 ip router isis 1
 ipv6 address A002::1/128
 ipv6 router isis 1
!
interface Loopback11
 vrf forwarding vpn-11
 ip address 11.11.11.11 255.255.255.255
 ip pim sparse-mode
!
interface Tunnel231
 description pe11_pe21_tunnel1
 ip address 23.11.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2311::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.1
 tunnel mode gre ipv6
 tunnel destination 2301::3
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
!
interface Tunnel232
 description pe11_p21_tunnel2
 ip address 23.12.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2312::2/64
 ipv6 router isis 1
```

```
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.2
 tunnel mode gre ipv6
 tunnel destination 2302::3
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-128
  !
!
interface Tunnel233
 description pe11_p21_tunnel3
 ip address 23.13.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2313::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.3
 tunnel mode gre ipv6
 tunnel destination 2303::3
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-129
  !
!
interface Tunnel234
 description pe11_p21_tunnel4
 ip address 23.14.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2314::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.4
 tunnel mode gre ipv6
 tunnel destination 2304::3
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-130
  !
!
interface Tunnel235
 description pe11_p21_tunnel5
 ip address 23.15.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2315::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.5
 tunnel mode gre ipv6
 tunnel destination 2305::3
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-131
```

```
  !
!
interface Tunnel236
 description pe11_p21_tunnel6
 ip address 23.16.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2316::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.6
 tunnel mode gre ipv6
 tunnel destination 2306::3
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-132
  !
!
interface Tunnel237
 description pe11_p21_tunnel7
 ip address 23.17.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2317::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.7
 tunnel mode gre ipv6
 tunnel destination 2307::3
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-133
  !
!
interface Tunnel238
 description pe11_p21_tunnel8
 ip address 23.18.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2318::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.8
 tunnel mode gre ipv6
 tunnel destination 2308::3
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-134
  !
!
interface Tunnel239
 description pe11_p21_tunnel9
 ip address 23.19.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2319::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
```

```
 tunnel source GigabitEthernet3.9
 tunnel mode gre ipv6
 tunnel destination 2309::3
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-135
  !
!
interface GigabitEthernet2.11
 encapsulation dot1Q 11
 vrf forwarding vpn-11
 ip address 12.1.0.2 255.255.0.0
 ip pim sparse-mode
!
interface GigabitEthernet3
 description pe11_pe21
 no ip address
 negotiation auto
!
interface GigabitEthernet3.1
 description pe11_pe21_subintf1
 encapsulation dot1Q 4001
 ip address 23.1.0.2 255.255.0.0
 ipv6 address 2301::2/64
!
interface GigabitEthernet3.2
 description pe11_pe21_subintf2
 encapsulation dot1Q 4002
 ip address 23.2.0.2 255.255.0.0
 ipv6 address 2302::2/64
!
interface GigabitEthernet3.3
 description pe11_pe21_subintf3
 encapsulation dot1Q 4003
 ip address 23.3.0.2 255.255.0.0
 ipv6 address 2303::2/64
!
interface GigabitEthernet3.4
 description pe11_pe21_subintf4
 encapsulation dot1Q 4004
 ip address 23.4.0.2 255.255.0.0
 ipv6 address 2304::2/64
!
interface GigabitEthernet3.5
 description pe11_pe21_subintf5
 encapsulation dot1Q 4005
 ip address 23.5.0.2 255.255.0.0
 ipv6 address 2305::2/64
!
interface GigabitEthernet3.6
 description pe11_pe21_subintf6
 encapsulation dot1Q 4006
 ip address 23.6.0.2 255.255.0.0
 ipv6 address 2306::2/64
!
interface GigabitEthernet3.7
 description pe11_pe21_subintf7
 encapsulation dot1Q 4007
 ip address 23.7.0.2 255.255.0.0
 ipv6 address 2307::2/64
!
interface GigabitEthernet3.8
```

```
 description pe11_pe21_subintf8
 encapsulation dot1Q 4008
 ip address 23.8.0.2 255.255.0.0
 ipv6 address 2308::2/64
!
interface GigabitEthernet3.9
 description pe11_pe21_subintf9
 encapsulation dot1Q 4009
 ip address 23.9.0.2 255.255.0.0
 ipv6 address 2309::2/64
!
interface GigabitEthernet4
 no ip address
 negotiation auto
!
interface GigabitEthernet5
 no ip address
 negotiation auto
!
interface GigabitEthernet6
 no ip address
 speed 1000
 no negotiation auto
!
interface GigabitEthernet7
 description pe11_pe12
 no ip address
 speed 1000
 no negotiation auto
!
interface GigabitEthernet7.1
 description pe11_pe12_subintf1
 encapsulation dot1Q 4011
 ip address 24.1.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2401::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis metric 10 level-1
 isis metric 10 level-2
 isis bfd
 isis ipv6 metric 10
 isis ipv6 bfd
!
interface GigabitEthernet7.2
 description pe11_pe12_subintf2
 encapsulation dot1Q 4012
 ip address 24.2.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2402::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-128
  !
!
interface GigabitEthernet7.3
 description pe11_pe12_subintf3
```

```
 encapsulation dot1Q 4013
 ip address 24.3.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2403::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-129
  !
!
interface GigabitEthernet7.4
 description pe11_pe12_subintf4
 encapsulation dot1Q 4014
 ip address 24.4.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2404::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-130
  !
!
interface GigabitEthernet7.5
 description pe11_pe12_subintf5
 encapsulation dot1Q 4015
 ip address 24.5.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2405::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-131
  !
!
interface GigabitEthernet7.6
 description pe11_pe12_subintf6
 encapsulation dot1Q 4016
 ip address 24.6.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2406::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-132
  !
!
interface GigabitEthernet7.7
 description pe11_pe12_subintf7
```

```
 encapsulation dot1Q 4017
 ip address 24.7.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2407::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-133
  !
!
interface GigabitEthernet7.8
 description pe11_pe12_subintf8
 encapsulation dot1Q 4018
 ip address 24.8.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2408::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-134
  !
!
interface GigabitEthernet7.9
 description pe11_pe12_subintf9
 encapsulation dot1Q 4019
 ip address 24.9.0.2 255.255.0.0
 ip router isis 1
 ipv6 address 2409::2/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-135
  !
!
segment-routing srv6
 sid holdtime 0
 encapsulation
  hop-limit propagate
  traffic-class propagate
 locators
  locator loc1-pe11
   prefix F:1:2::/48
   format usid-f3216
  locator F128
   prefix F:F128:2::/48
   format usid-f3216
   algorithm 128
  locator F129
   prefix F:F129:2::/48
   format usid-f3216
   algorithm 129
  locator F130
```

```
   prefix F:F130:2::/48
   format usid-f3216
   algorithm 130
  locator F131
   prefix F:F131:2::/48
   format usid-f3216
   algorithm 131
  locator F132
   prefix F:F132:2::/48
   format usid-f3216
   algorithm 132
  locator F133
   prefix F:F133:2::/48
   format usid-f3216
   algorithm 133
  locator F134
   prefix F:F134:2::/48
   format usid-f3216
   algorithm 134
  locator F135
   prefix F:F135:2::/48
   format usid-f3216
   algorithm 135
!
router isis 1
 net 49.0001.0000.0000.0002.00
 is-type level-2-only
 router-id Loopback0
 metric-style wide
 log-adjacency-changes
 lsp-mtu 1418
 nsf ietf
 distribute link-state level-2
 segment-routing mpls
 segment-routing prefix-sid-map advertise-local
 fast-reroute per-prefix level-2 all
 fast-reroute remote-lfa level-2 mpls-ldp
 fast-reroute ti-lfa level-2
 microloop avoidance rib-update-delay 15000
 affinity-map flexalgo-128 bit-position 128
 affinity-map flexalgo-129 bit-position 129
 affinity-map flexalgo-130 bit-position 130
 affinity-map flexalgo-131 bit-position 131
 affinity-map flexalgo-132 bit-position 132
 affinity-map flexalgo-133 bit-position 133
 affinity-map flexalgo-134 bit-position 134
 affinity-map flexalgo-135 bit-position 135
 flex-algo 128
  advertise-definition
  metric-type delay
  priority 100
  affinity
   include-any
    name flexalgo-128
    exit-fa-affinity-attr
   !
 !
 flex-algo 129
  advertise-definition
  metric-type delay
  priority 100
  affinity
   include-any
    name flexalgo-129
```

```
   exit-fa-affinity-attr
  !
!
flex-algo 130
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-130
   exit-fa-affinity-attr
  !
!
flex-algo 131
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-131
   exit-fa-affinity-attr
  !
!
flex-algo 132
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-132
   exit-fa-affinity-attr
  !
!
flex-algo 133
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-133
   exit-fa-affinity-attr
  !
!
flex-algo 134
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-134
   exit-fa-affinity-attr
  !
!
flex-algo 135
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-135
   exit-fa-affinity-attr
  !
!
bfd all-interfaces
```

```
 !
 address-family ipv6
  bfd all-interfaces
  segment-routing srv6
   locator loc1-pe11
   locator F128
   locator F129
   locator F130
   locator F131
   locator F132
   locator F133
   locator F134
   locator F135
  fast-reroute per-prefix level-2 all
  fast-reroute ti-lfa level-2
  microloop avoidance segment-routing
  microloop avoidance rib-update-delay 15000
 exit-address-family
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-2
!
router bgp 100
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor A005::1 remote-as 100
 neighbor A005::1 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv4 mvpn
  neighbor A005::1 activate
  neighbor A005::1 send-community extended
 exit-address-family
 !
 address-family vpnv4
  neighbor A005::1 activate
  neighbor A005::1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf vpn-11
  redistribute connected
  redistribute static
  !
  segment-routing srv6
   locator F128
   alloc-mode per-vrf
  exit-srv6
  !
 exit-address-family
 !
!
ip pim vrf vpn-11 autorp listener
ip pim vrf vpn-11 send-rp-announce Loopback11 scope 10
ip pim vrf vpn-11 send-rp-discovery Loopback11 scope 10
ip pim vrf vpn-11 ssm default


ip route vrf vpn-11 11.1.0.0 255.255.0.0 GigabitEthernet2.11 12.1.0.1

performance-measurement
 interface GigabitEthernet7.1
```

```
 delay-measurement
  advertise-delay 10
interface GigabitEthernet7.2
 delay-measurement
  advertise-delay 10
interface GigabitEthernet7.3
 delay-measurement
  advertise-delay 10
interface GigabitEthernet7.4
 delay-measurement
  advertise-delay 10
interface GigabitEthernet7.5
 delay-measurement
  advertise-delay 10
interface GigabitEthernet7.6
 delay-measurement
  advertise-delay 10
interface GigabitEthernet7.7
 delay-measurement
  advertise-delay 10
interface GigabitEthernet7.8
 delay-measurement
  advertise-delay 10
interface GigabitEthernet7.9
 delay-measurement
  advertise-delay 10
interface Tunnel231
 delay-measurement
  advertise-delay 10
interface Tunnel232
 delay-measurement
  advertise-delay 10
interface Tunnel233
 delay-measurement
  advertise-delay 10
interface Tunnel234
 delay-measurement
  advertise-delay 10
interface Tunnel235
 delay-measurement
  advertise-delay 10
interface Tunnel236
 delay-measurement
  advertise-delay 10
interface Tunnel237
 delay-measurement
  advertise-delay 10
interface Tunnel238
 delay-measurement
  advertise-delay 10
interface Tunnel239
 delay-measurement
  advertise-delay 10
!
```

SRv6 configuration for PE12

This section details the configuration of PE12, a Provider Edge router, focusing on its role in an SRv6 enabled network with advanced traffic engineering and VPN services.

```
version 17.19
hostname PE12
```

```
vrf definition vpn-11
 rd 11:11
 route-target export 11:11
 route-target import 11:11
 !
 address-family ipv4
  srv6-mcast ingress-replication partitioned data
  route-target export 11:11
  route-target import 11:11
 exit-address-family
 !
 address-family ipv6
  route-target export 11:11
  route-target import 11:11
 exit-address-family
!
ip multicast-routing vrf vpn-11 distributed

bfd-template single-hop bfd_tmp
 interval microseconds min-tx 7000 min-rx 7000 multiplier 3
!
crypto ikev2 keyring gre_key
 peer gre
  address ::/0
  pre-shared-key cisco123
 !
!
!
crypto ikev2 profile gre_profile
 match identity remote address ::/0
 authentication remote pre-share
 authentication local pre-share
 keyring local gre_key

crypto ipsec transform-set gre_transet esp-aes esp-sha-hmac
 mode transport
!
!
crypto ipsec profile gre_profile
 set transform-set gre_transet
 set ikev2-profile gre_profile
!
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
 ip router isis 1
 ipv6 address A004::1/128
 ipv6 router isis 1
!
interface Tunnel451
 ip address 45.11.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 4511::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.1
 tunnel mode gre ipv6
 tunnel destination 4501::5
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
!
interface Tunnel452
```

```
 description pe12_p_tunnel2
 ip address 45.12.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 4512::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.2
 tunnel mode gre ipv6
 tunnel destination 4502::5
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-128
  !
!
interface Tunnel453
 description pe12_p_tunnel3
 ip address 45.13.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 4513::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.3
 tunnel mode gre ipv6
 tunnel destination 4503::5
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-129
  !
!
interface Tunnel454
 description pe12_p_tunnel4
 ip address 45.14.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 4514::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.4
 tunnel mode gre ipv6
 tunnel destination 4504::5
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-130
  !
!
interface Tunnel455
 description pe12_p_tunnel5
 ip address 45.15.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 4515::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.5
 tunnel mode gre ipv6
 tunnel destination 4505::5
```

```
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-131
  !
!
interface Tunnel456
 description pe12_p_tunnel6
 ip address 45.16.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 4516::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.6
 tunnel mode gre ipv6
 tunnel destination 4506::5
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-132
  !
!
interface Tunnel457
 description pe12_p_tunnel7
 ip address 45.17.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 4517::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.7
 tunnel mode gre ipv6
 tunnel destination 4507::5
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-133
  !
!
interface Tunnel458
 description pe12_p_tunnel8
 ip address 45.18.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 4518::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.8
 tunnel mode gre ipv6
 tunnel destination 4508::5
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-134
  !
!
interface Tunnel459
 description pe12_p_tunnel9
 ip address 45.19.0.4 255.255.0.0
```

```
 ip router isis 1
 ipv6 address 4519::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 tunnel source GigabitEthernet3.9
 tunnel mode gre ipv6
 tunnel destination 4509::5
 tunnel path-mtu-discovery
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-135
  !
!
interface GigabitEthernet2
 description pe12_fhr
 vrf forwarding red
 ip address 14.1.0.4 255.255.0.0
 ip pim sparse-mode
 negotiation auto
!
interface GigabitEthernet2.11
 encapsulation dot1Q 11
 vrf forwarding vpn-11
 ip address 14.1.0.4 255.255.0.0
 ip pim sparse-mode
!
interface GigabitEthernet3
 description pe12_p
 no ip address
 negotiation auto
!
interface GigabitEthernet3.1
 description pe12_p_subintf1
 encapsulation dot1Q 4031
 ip address 45.1.0.4 255.255.0.0
 ipv6 address 4501::4/64
!
interface GigabitEthernet3.2
 description pe12_p_subintf2
 encapsulation dot1Q 4032
 ip address 45.2.0.4 255.255.0.0
 ipv6 address 4502::4/64
!
interface GigabitEthernet3.3
 description pe12_p_subintf3
 encapsulation dot1Q 4033
 ip address 45.3.0.4 255.255.0.0
 ipv6 address 4503::4/64
!
interface GigabitEthernet3.4
 description pe12_p_subintf4
 encapsulation dot1Q 4034
 ip address 45.4.0.4 255.255.0.0
 ipv6 address 4504::4/64
!
interface GigabitEthernet3.5
 description pe12_p_subintf5
 encapsulation dot1Q 4035
 ip address 45.5.0.4 255.255.0.0
 ipv6 address 4505::4/64
!
interface GigabitEthernet3.6
```

```
 description pe12_p_subintf6
 encapsulation dot1Q 4036
 ip address 45.6.0.4 255.255.0.0
 ipv6 address 4506::4/64
!
interface GigabitEthernet3.7
 description pe12_p_subintf7
 encapsulation dot1Q 4037
 ip address 45.7.0.4 255.255.0.0
 ipv6 address 4507::4/64
!
interface GigabitEthernet3.8
 description pe12_p_subintf8
 encapsulation dot1Q 4038
 ip address 45.8.0.4 255.255.0.0
 ipv6 address 4508::4/64
!
interface GigabitEthernet3.9
 description pe12_p_subintf9
 encapsulation dot1Q 4039
 ip address 45.9.0.4 255.255.0.0
 ipv6 address 4509::4/64
!
interface GigabitEthernet5
 description pe12_p11
 no ip address
 speed 1000
 no negotiation auto
!
interface GigabitEthernet5.1
 description pe12_p11_subintf1
 encapsulation dot1Q 4011
 ip address 24.1.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 2401::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
!
interface GigabitEthernet5.2
 description pe12_p11_subintf2
 encapsulation dot1Q 4012
 ip address 24.2.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 2402::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-128
  !
!
interface GigabitEthernet5.3
 description pe12_p11_subintf3
 encapsulation dot1Q 4013
 ip address 24.3.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 2403::4/64
```

```
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-129
  !
!
interface GigabitEthernet5.4
 description pe12_p11_subintf4
 encapsulation dot1Q 4014
 ip address 24.4.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 2404::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-130
  !
!
interface GigabitEthernet5.5
 description pe12_p11_subintf5
 encapsulation dot1Q 4015
 ip address 24.5.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 2405::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-131
  !
!
interface GigabitEthernet5.6
 description pe12_p11_subintf6
 encapsulation dot1Q 4016
 ip address 24.6.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 2406::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-132
  !
!
interface GigabitEthernet5.7
 description pe12_p11_subintf7
 encapsulation dot1Q 4017
 ip address 24.7.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 2407::4/64
```

```
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-133
  !
!
interface GigabitEthernet5.8
 description pe12_p11_subintf8
 encapsulation dot1Q 4018
 ip address 24.8.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 2408::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-134
  !
!
interface GigabitEthernet5.9
 description pe12_p11_subintf9
 encapsulation dot1Q 4019
 ip address 24.9.0.4 255.255.0.0
 ip router isis 1
 ipv6 address 2409::4/64
 ipv6 router isis 1
 mpls traffic-eng tunnels
 bfd template bfd_tmp
 isis network point-to-point
 isis bfd
 isis ipv6 bfd
 isis affinity flex-algo
  name flexalgo-135
  !
!
segment-routing srv6
 sid holdtime 0
 encapsulation
  hop-limit propagate
  traffic-class propagate
 locators
  locator loc1-pe12
   prefix F:1:4::/48
   format usid-f3216
  locator F128
   prefix F:F128:4::/48
   format usid-f3216
   algorithm 128
  locator F129
   prefix F:F129:4::/48
   format usid-f3216
   algorithm 129
  locator F130
   prefix F:F130:4::/48
   format usid-f3216
   algorithm 130
  locator F131
```

```
   prefix F:F131:4::/48
   format usid-f3216
   algorithm 131
  locator F132
   prefix F:F132:4::/48
   format usid-f3216
   algorithm 132
  locator F133
   prefix F:F133:4::/48
   format usid-f3216
   algorithm 133
  locator F134
   prefix F:F134:4::/48
   format usid-f3216
   algorithm 134
  locator F135
   prefix F:F135:4::/48
   format usid-f3216
   algorithm 135
!
router isis 1
 net 49.0001.0000.0000.0004.00
 is-type level-2-only
 router-id Loopback0
 metric-style wide
 log-adjacency-changes
 lsp-mtu 1418
 nsf ietf
 distribute link-state level-2
 segment-routing mpls
 segment-routing prefix-sid-map advertise-local
 fast-reroute per-prefix level-2 all
 fast-reroute remote-lfa level-2 mpls-ldp
 fast-reroute ti-lfa level-2
 microloop avoidance rib-update-delay 15000
 affinity-map flexalgo-128 bit-position 128
 affinity-map flexalgo-129 bit-position 129
 affinity-map flexalgo-130 bit-position 130
 affinity-map flexalgo-131 bit-position 131
 affinity-map flexalgo-132 bit-position 132
 affinity-map flexalgo-133 bit-position 133
 affinity-map flexalgo-134 bit-position 134
 affinity-map flexalgo-135 bit-position 135
 flex-algo 128
  advertise-definition
  metric-type delay
  priority 100
  affinity
   include-any
    name flexalgo-128
    exit-fa-affinity-attr
   !
 !
 flex-algo 129
  advertise-definition
  metric-type delay
  priority 100
  affinity
   include-any
    name flexalgo-129
    exit-fa-affinity-attr
   !
 !
 flex-algo 130
```

```
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-130
   exit-fa-affinity-attr
  !
!
flex-algo 131
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-131
   exit-fa-affinity-attr
  !
!
flex-algo 132
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-132
   exit-fa-affinity-attr
  !
!
flex-algo 133
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-133
   exit-fa-affinity-attr
  !
!
flex-algo 134
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-134
   exit-fa-affinity-attr
  !
!
flex-algo 135
 advertise-definition
 metric-type delay
 priority 100
 affinity
  include-any
   name flexalgo-135
   exit-fa-affinity-attr
  !
!
bfd all-interfaces
!
address-family ipv6
 bfd all-interfaces
 segment-routing srv6
```

```
   locator loc1-pe12
   locator F128
   locator F129
   locator F130
   locator F131
   locator F132
   locator F133
   locator F134
   locator F135
  fast-reroute per-prefix level-2 all
  fast-reroute ti-lfa level-2
  microloop avoidance segment-routing
  microloop avoidance rib-update-delay 15000
 exit-address-family
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-2
!
router bgp 100
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor A005::1 remote-as 100
 neighbor A005::1 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv4 mvpn
  neighbor A005::1 activate
  neighbor A005::1 send-community extended
 exit-address-family
 !
 address-family vpnv4
  neighbor A005::1 activate
  neighbor A005::1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf red
  redistribute connected
  redistribute static
  !
  segment-routing srv6
   locator loc1-pe12
   alloc-mode per-vrf
  exit-srv6
  !
 exit-address-family
 !
 address-family ipv4 vrf vpn-11
  redistribute connected
  redistribute static
  !
  segment-routing srv6
   locator F128
   alloc-mode per-vrf
  exit-srv6
  !
 exit-address-family
 !
ip pim vrf vpn-11 autorp listener
ip pim vrf vpn-11 ssm default

ip route vrf vpn-11 11.1.0.0 255.255.0.0 GigabitEthernet2.11 14.1.0.1
```

```
performance-measurement
 interface GigabitEthernet5.1
  delay-measurement
   advertise-delay 10
 interface GigabitEthernet5.2
  delay-measurement
   advertise-delay 10
 interface GigabitEthernet5.3
  delay-measurement
   advertise-delay 10
 interface GigabitEthernet5.4
  delay-measurement
   advertise-delay 10
 interface GigabitEthernet5.5
  delay-measurement
   advertise-delay 10
 interface GigabitEthernet5.6
  delay-measurement
   advertise-delay 10
 interface GigabitEthernet5.7
  delay-measurement
   advertise-delay 10
 interface GigabitEthernet5.8
  delay-measurement
   advertise-delay 10
 interface GigabitEthernet5.9
  delay-measurement
   advertise-delay 10
 interface Tunnel451
  delay-measurement
   advertise-delay 10
 interface Tunnel452
  delay-measurement
   advertise-delay 10
 interface Tunnel453
  delay-measurement
   advertise-delay 10
 interface Tunnel454
  delay-measurement
   advertise-delay 10
 interface Tunnel455
  delay-measurement
   advertise-delay 10
 interface Tunnel456
  delay-measurement
   advertise-delay 10
 interface Tunnel457
  delay-measurement
   advertise-delay 10
 interface Tunnel458
  delay-measurement
   advertise-delay 10
 interface Tunnel459
  delay-measurement
   advertise-delay 10
```