



JUNOS™
Internet Software
Configuration Guide
MPLS Applications

Release 5.0

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Part Number: 530-004547-01, Revision 1

Juniper
Networks

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986–1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.BSD and 4.BSD-Lite Releases is copyrighted by The Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, The Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., Copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks is a registered trademark of Juniper Networks, Inc. Internet Processor, Internet Processor II, JUNOS, JUNOScript, M5, M10, M20, M40, and M160 are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks may be the property of their respective owners. All specifications are subject to change without notice.

JUNOS Internet Software Configuration Guide: MPLS Applications, Release 5.0

Copyright © 2001, Juniper Networks, Inc.
All rights reserved. Printed in USA.

Writer: Albert Statti
Editor: Pam Muraca, Cris Morris
Covers and template design: Edmonds Design

Revision History
10 August 2001—First edition.

The information in this document is current as of the date listed in the revision history above.

The information in this document has been carefully verified and is believed to be accurate. Juniper Networks assumes no responsibilities for any inaccuracies that may appear in this document. In no event will Juniper Networks be liable for direct, indirect, special, exemplary, incidental or consequential damages resulting from any defect or omission in this document, even if advised of the possibility of such damages.

Juniper Networks reserves the right to change, modify, transfer or otherwise revise this publication without notice.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The JUNOS software has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

Table of Contents

About This Manual

Objectives	xix
Audience	xx
Document Organization	xx
Related Documentation	xxii
Manual Part Organization	xxiii
Using the Indexes	xxiv
Documentation Conventions	xxiv
General Conventions	xxiv
Conventions for Software Commands and Statements	xxv
Documentation Feedback	xxvi
How to Request Support	xxvi

Part 1

Overview

Chapter 1

Traffic Engineering Overview	3
------------------------------------	---

Components of Traffic Engineering	3
Packet Forwarding Component	4
Packet Forwarding Based on Label Swapping	4
Example of How a Packet Traverses an MPLS Backbone	4
Information Distribution Component	5
Path Selection Component	5
Offline Planning and Analysis	6
Signaling Component	6
Flexible LSP Calculation and Configuration	6

Chapter 2

Complete MPLS Applications

Configuration Mode Statements 9

[edit protocols connections] Hierarchy Level 10

[edit protocols l2vpn] Hierarchy Level 10

[edit protocols ldp] Hierarchy Level 11

[edit protocols mpls] Hierarchy Level 11

[edit protocols rsvp] Hierarchy Level 13

[edit routing-instances] Hierarchy Level (for Layer 2 VPNs) 14

[edit routing-instances] Hierarchy Level (for Layer 3 VPNs) 14

Part 2 MPLS

Chapter 3

MPLS Overview 19

MPLS Standards..... 19

Link-Layer Support 20

MPLS and Traffic Engineering..... 20

Label Description..... 21

Special Labels..... 21

Label Allocation..... 22

Operations on Labels 23

Routers in an LSP 24

How a Packet Travels along an LSP 24

Types of LSPs 24

Scope of LSPs 25

Constrained-Path LSP Computation..... 25

How CSPF Selects a Path..... 26

Path Selection Tie-Breaking..... 27

Computing Paths Offline 27

Fate Sharing 28

IGP Shortcuts..... 28

Enable IGP Shortcuts..... 30

LSPs Qualified in Shortcut Computations 30

IGP Shortcut Applications..... 30

IGP Shortcuts and Routing Table 31

Router Requirements 31

Advertise LSPs into IGP..... 32

MPLS Applications 33

BGP Destinations..... 33

IGP and BGP Destinations..... 35

Select Forwarding LSP Next Hop 35

MPLS and Routing Tables 36

MPLS and Traffic Protection 38

Per-Prefix Load Balancing..... 38

Chapter 4

MPLS Configuration Statements39

Minimum MPLS Configuration41

Chapter 5

Configure MPLS Signaled LSPs43

Configure the Ingress Router for Signaled LSPs43

Create a Named Path44

Examples: Create a Named Path45

Create an LSP45

Configure the Address of the Egress Router48

Configure the Address of the Ingress Router48

Configure the Primary and Secondary LSPs49

Configure Fast Reroute49

Configure Addresses to Associate with the LSP52

Configure Path Connection Retry Information53

Configure the Dynamic LSP Metric53

Configure the Static LSP Metric54

Configure CSPF Tie Breaking54

Configure Load-Balancing LSPs without CSPF55

Disable Normal TTL Decrementing55

Disable Constrained Path LSP Computation56

Configure Administrative Groups57

Configure the LSP Preference59

Configure Whether to Record Path Routes59

Configure the MPLS CoS Value60

Configure an LSP to Be Adaptive61

Configure Priority and Preemption62

Optimize Signaled LSPs63

Configure the Maximum Path Length64

Configure the Path Bandwidth64

Configure the Standby State64

Configure LSP Hold Time65

Configure LDP Tunneling65

Configure Alternate Backup Paths Using Fate Sharing66

Implications to CSPF67

Example: Configure Fate Sharing67

Configure All Other MPLS Routers for Signaled LSPs68

Enable RSVP68

Examples: Configure Signaled LSPs68

Configure MPLS over GRE Tunnels71

Example: Configure MPLS over GRE Tunnels71

Chapter 6	Configure Static LSPs	73
	Configure the Ingress Router for Static MPLS.....	73
	Example: Configure the Ingress Router	74
	Configure the Intermediate and Egress Routers for Static MPLS	75
	Example: Configure an Intermediate Router.....	77
	Example: Configure an Egress Router.....	77
Chapter 7	Configure Explicit-Path LSPs	79
Chapter 8	Configure Miscellaneous MPLS Properties.....	81
	Configure Traffic Engineering for LSPs	81
	Configure MPLS to Gather Statistics	81
	Control MPLS System Log Messages and SNMP Traps	82
	Trace MPLS Protocol Packets and Operations	83
Chapter 9	Summary of MPLS Configuration Statements.....	85
	adaptive.....	85
	admin-group.....	85
	admin-groups	86
	advertise-hold-time.....	86
	bandwidth	87
	class-of-service	87
	disable	88
	discard	88
	exclude	88
	exclude (for administrative groups)	88
	exclude (for fast reroute)	89
	fast-reroute	89
	fate-sharing.....	89
	from	90
	hop-limit.....	91
	include	91
	include (for administrative groups).....	91
	include (for fast reroute)	92
	install.....	92
	interface	92
	label-map.....	93
	label-switched-path.....	94
	ldp-tunneling	95
	least-fill	95
	log-updown.....	95
	metric.....	96

most-fill	96
mpls	96
nexthop	96
no-cspf	97
no-decrement-ttl	97
no-exclude	98
no-include	98
no-propagate-ttl	98
no-record	98
optimize-aggressive	99
optimize-timer	99
path	100
pop	100
preference	101
primary	101
priority	102
push	102
random	103
record	103
reject	104
retry-limit	104
retry-timer	104
secondary	105
standby	105
static-path	106
statistics	106
swap	107
to	108
traceoptions	108
traffic-engineering	110
type	111

Part 3 RSVP

Chapter 10

RSVP Overview	115
---------------------	-----

RSVP Overview	115
RSVP Standards	116
JUNOS RSVP Protocol Implementation	116
RSVP Operation	117
RSVP Message Types	117
Path Messages	117
Resv Messages	118
PathTear Messages	118
ResvTear Messages	118
PathErr Messages	118
ResvErr Messages	118
ResvConfirm Messages	119
RSVP Reservation Styles	119

Chapter 11

RSVP Configuration Guidelines..... 121

Minimum RSVP Configuration	122
Enable RSVP	122
Configure RSVP Aggregation.....	122
Configure the RSVP Hello Interval.....	123
Configure RSVP Authentication.....	123
Reserve Bandwidth on an Interface	124
Configure RSVP Timers.....	124
Preempt RSVP Sessions	125
Trace RSVP Protocol Traffic	125
Examples: Trace RSVP Protocol Traffic.....	126
Configure RSVP and MPLS.....	127
Example: Configure RSVP and MPLS	127

Chapter 12

Summary of RSVP Configuration Statements..... 129

aggregate.....	129
authentication-key	130
bandwidth	130
disable	131
hello-interval.....	131
interface	131
keep-multiplier	132
no-aggregate.....	132
preemption.....	132
refresh-time	133
rsvp	133
subscription	134
traceoptions.....	134

Part 4

LDP

Chapter 13

LDP Overview..... 139

Overview	139
LDP Standards	140
JUNOS LDP Protocol Implementation	140
LDP Operation	140
LDP Label Filtering.....	140
Tunneling LDP LSPs in RSVP LSPs	141
Label Operations	141
Restrictions for LDP over RSVP	142
LDP Message Types	142
Discovery Messages.....	142
Session Messages	142
Advertisement Messages	143
Notification Messages	143

Chapter 14

Configure LDP..... 145

Minimum LDP Configuration	146
Enable LDP	146
Configure the LDP Hello Interval.....	147
Configure the LDP Hold Time	147
Configure the LDP Keepalive Interval.....	147
Configure the LDP Keepalive Timeout.....	147
Configure LDP Route Preferences	148
Configure LDP Received Label Filtering	148
Examples: Configure Received Label Filtering	149
Configure LDP Outbound Label Filtering.....	150
Examples: Configure Outbound Label Filtering.....	151
Enable LDP over RSVP-Established LSPs.....	152
Configure LDP Transport Address Control	152
Configure LDP Egress Policy	152
Examples: Configure Egress Policy	153
Configure FEC Deaggregation	153
Trace LDP Protocol Traffic	154
Examples: Trace LDP Protocol Traffic	155
Example: LDP Configuration.....	155

Chapter 15

Summary of LDP Configuration Statements..... 157

deaggregate	157
disable	157
egress-policy	158
export	158
hello-interval	158
hold-time	159
import	159
interface	159
keepalive-interval	160
keepalive-timeout	160
ldp	160
no-deaggregate	161
preference	161
traceoptions	161
transport-address	163

Part 5 CCC

Chapter 16

CCC Overview..... 167

Chapter 17

CCC Configuration..... 169

Configure Layer 2 Switching Cross-Connects	169
Define the CCC Encapsulation for Layer 2 Switching Cross-Connects	170
Define the CCC Connection for Layer 2 Switching Cross-Connects	171
Configure MPLS	171
Example: Configure Layer 2 Switching Cross-Connects	172
Configure MPLS LSP Tunnel Cross-Connects	173
Define the CCC Encapsulation for LSP Tunnel Cross-Connects	174
Define the CCC Connection for LSP Tunnel Cross-Connects	175
Example: Configure LSP Tunnel Cross-Connects	176
Configure LSP Stitching Cross-Connects	177
Example: Configure LSP Stitching Cross-Connects	178

Chapter 18

Summary of CCC Configuration Statements..... 179

connections	179
interface-switch	180
lsp-switch	180
remote-interface-switch	181

Part 6

VPNs

Chapter 19

Layer 3 VPN Overview.....185

Layer 3 VPN Overview	185
Layer 3 VPN Standards	186
VPN Terminology.....	186
VPN Attributes	187
VPN-IPv4 Addresses and Route Distinguishers.....	188
VPN Routing and Forwarding Tables.....	191
Route Distribution within a VPN	194
Distribution of Routes from CE to PE Routers.....	195
Distribution of Routes between PE Routers	196
Distribution of Routes from PE to CE Routers.....	197
Forwarding across the Provider's Core Network	198
Routing Instances for VPNs.....	200

Chapter 20

Layer 3 VPN Configuration Guidelines.....201

Enable a Signaling Protocol	203
Use LDP for VPN Signaling	203
Use RSVP for VPN Signaling	204
Configure an IGP on PE and Provider Routers.....	206
Configure an IBGP Session between PE Routers	206
Configure Routing Instances for VPNs on PE Routers	207
Configure the Instance Type	207
Configure Interfaces for VPN Routing	207
Configure the Route Distinguisher	208
Configure Policy for the PE Router's VRF Table	209
Configure the Route Target.....	209
Configure the Route Origin.....	210
Configure Import Policy for the PE Router's VRF Table	210
Configure Export Policy for the PE Router's VRF Table	211
Configure VPN Routing between the PE and CE Routers.....	213
Configure BGP between the PE and CE Routers.....	213
Configure OSPF between the PE and CE routers.....	213
Configure RIP between the PE and CE Routers.....	214
Configure Static Routes between the PE and CE Routers	214

Chapter 21

Layer 3 VPN Configuration Troubleshooting Guidelines..... 215

Diagnose Common Problems 215

Use the Ping and Traceroute Commands to Troubleshoot

Layer 3 VPN Topologies..... 219

Ping One CE Router from the Other 220

Ping the Remote PE and CE Routers from the Local CE Router 221

Ping the Directly Connected PE and CE Routers from Each Other 223

Chapter 22

Layer 3 VPN Configuration Examples 227

Configure a Simple Full-Mesh VPN Topology 227

Enable an IGP on the PE and Provider Routers..... 229

Enable RSVP and MPLS on the Provider Router..... 229

Configure the MPLS LSP Tunnel between the PE Routers 230

Configure IBGP on the PE Routers 231

Configure Routing Instances for VPNs on the PE Routers 232

Configure VPN Policy on the PE Routers 234

Simple VPN Configuration Summarized by Router 237

Router A (PE Router) 237

Router B (Provider Router) 239

Router C (PE Router) 240

Configure a Full-Mesh VPN Topology with Route Reflectors..... 242

Configure a Hub-and-Spoke VPN Topology 242

Enable an IGP on the Hub and Spoke PE Routers 245

Configure LDP on the Hub and Spoke PE Routers 245

Configure IBGP on the PE Routers 246

Configure Routing Instances for VPNs on the Hub and Spoke PE Routers ... 247

Configure VPN Policy on the PE Routers 249

Hub-and-Spoke VPN Configuration Summarized by Router 252

Router D (Hub PE Router) 252

Router E (Spoke PE Router) 254

Router F (Spoke PE Router) 256

Configure an LDP-over-RSVP VPN Topology 257

Enable an IGP on the PE and Provider Routers..... 261

Enable LDP on the PE and Provider Routers..... 261

Enable RSVP and MPLS on the Provider Router..... 263

Configure the MPLS LSP Tunnel between the Provider Routers 263

Configure IBGP on the PE Routers 264

Configure Routing Instances for VPNs on the PE Routers 265

Configure VPN Policy on the PE Routers 266

LDP-over-MPLS VPN Configuration Summarized by Router 268

Router PE1 268

Router P1 269

Router P2 270

Router P3 270

Router PE2 271

Configure an Application-Based Layer 3 VPN Topology 272

Configuration on Router A..... 274

Configuration on Router E..... 276

Configuration for Router F..... 277

Chapter 23

Summary of Layer 3 VPN Configuration Statements279

instance-type	279
interface	279
route-distinguisher	280
vrf-export	280
vrf-import	280

Chapter 24

Layer 2 VPN Overview.....281

Layer 2 VPN Overview	281
Layer 2 VPN Standards	282

Chapter 25

Layer 2 VPN Configuration Guidelines.....283

Configure MPLS LSPs between the PE Routers	284
Configure MPLS LSPs using LDP	285
Configure MPLS LSPs Using RSVP	286
Configure an IGP on PE and Provider Routers	288
Configure an IBGP Session between PE Routers	288
Configure Routing Instances for Layer 2 VPNs on the PE Routers	289
Configure the Instance Type	289
Configure Interfaces for Layer 2 VPN Routing.....	289
Configure CCC Encapsulation on Interfaces	290
Configure the Route Distinguisher	291
Configure Policy for the PE Router's VRF Table	291
Configure the Connections to the Local Site.....	291
Configure the Local Site	292
Configure the Encapsulation Type	292
Examine Layer 2 VPN Traffic Using Trace Options	293

Chapter 26

Layer 2 VPN Configuration Example295

Enable an IGP on the PE routers	296
Configure MPLS LSP Tunnels between the PE Routers	296
Configure IBGP on the PE Routers and Provider Routers	298
Configure Routing Instances for Layer 2 VPNs on the PE Routers	299
Configure VPN Policy on the PE Routers.....	302
Layer 2 VPN Configuration Summarized by Router.....	304
Summary for Router A (PE Router for Sunnyvale)	304
Summary for Router B (PE Router for Austin)	306
Summary for Router C (PE Router for Portland)	308

Chapter 27

Summary of Layer 2 VPN Configuration Statements.....	311
--	-----

encapsulation-type	311
encapsulation-type (ccc)	311
encapsulation-type (layer 2 vpn).....	312
instance-type	312
interface	313
route-distinguisher	313
site	314
traceoptions	314
vrf-export	316
vrf-import	316

Part 7

Appendix

Appendix A

Glossary	319
----------------	-----

Part 8

Indexes

Index

Comprehensive Index

Index

Comprehensive Index of Statements and Commands

List of Figures

List of Figures

Figure 1:	Label Encoding	23
Figure 2:	CSPF Computation Process	26
Figure 3:	Typical SPF Tree, Sourced from Router A	29
Figure 4:	Modified SPF Tree, Using LSP A–D as a Shortcut	29
Figure 5:	Modified SPF Tree, Using Both LSP A–D and LSP A–E as Shortcuts	30
Figure 6:	IGP Shortcuts	31
Figure 7:	IGP Shortcuts in a Bigger Network	31
Figure 8:	SPF Computations with Advertised LSPs	32
Figure 9:	MPLS Application Topology	34
Figure 10:	How BGP Determines How to Reach Next-Hop Addresses	35
Figure 11:	MPLS Routing and Forwarding Tables When traffic-engineering bgp Is Configured	36
Figure 12:	MPLS Routing and Forwarding Tables When traffic-engineering bgp-igp Is Configured	37
Figure 13:	Detours Established for an LSP Using Fast Reroute	50
Figure 14:	Detour after the Link from Router B to Router C Fails	50
Figure 15:	Detours Merging into Other Detours	51
Figure 16:	Static MPLS Configuration	75
Figure 17:	Swap and Push Label Operation When Tunneling LDP LSPs through RSVP LSPs	141
Figure 18:	Double Push Label Operation When Tunneling LDP LSPs through RSVP LSPs	142
Figure 19:	Layer 2 Switching Cross-Connect	169
Figure 20:	Sample Topology of Frame Relay Layer 2 Switching Cross-Connect	172
Figure 21:	Sample Topology of a VLAN Layer 2 Switching Cross-Connect	172
Figure 22:	MPLS LSP Tunnel Cross-Connect	173
Figure 23:	Example Topology of MPLS LSP Tunnel Cross-Connect	176
Figure 24:	LSP Stitching Cross-Connect	177
Figure 25:	Example Topology of LSP Stitching Cross-Connect	178
Figure 26:	VPN Router Components	187
Figure 27:	VPN Attributes and Route Distribution	188
Figure 28:	Overlapping Addresses among Different VPNs	189
Figure 29:	VPN-IPv4 Address Format	190
Figure 30:	Route Distinguishers	191
Figure 31:	VRF Tables	192
Figure 32:	Route Distribution within a VPN	194
Figure 33:	Distribution of Routes from CE Routers to PE Routers	196
Figure 34:	Distribution of Routes between PE Routers	197
Figure 35:	Distribution of Routes from PE Routers to CE Routers	198
Figure 36:	Using MPLS LSPs to Tunnel between PE Routers	199
Figure 37:	Label Stack	199
Figure 38:	Layer 3 VPN Topology for Ping and Traceroute Command Examples	219
Figure 39:	Example of a Simple VPN Topology	228

List of Figures

Figure 40: Example of a Hub-and-Spoke VPN Topology.....	243
Figure 41: Route Distribution between Two Spoke Routers	244
Figure 42: Example of an LDP-over-RSVP VPN Topology.....	258
Figure 43: Label Pushing and Popping.....	260
Figure 44: Application-Based Layer 3 VPN Example Configuration	274
Figure 45: Layer 2 VPN Connecting CE Routers.....	282
Figure 46: Example of a Simple Layer 2 VPN Topology	295

List of Tables

List of Tables

Table 1:	MPLS CoS Values	61
Table 2:	from Operators That Apply to LDP Received Label Filtering	148
Table 3:	to Operators for LDP Outbound Label Filtering	150

List of Tables

About This Manual

This chapter provides a high-level overview of the *JUNOS Internet Software Configuration Guide: MPLS Applications*:

- Objectives on page xix
- Audience on page xx
- Document Organization on page xx
- Related Documentation on page xxii
- Manual Part Organization on page xxiii
- Using the Indexes on page xxiv
- Documentation Conventions on page xxiv
- Documentation Feedback on page xxvi
- How to Request Support on page xxvi

Objectives

This manual provides an overview of the MPLS applications functions of the JUNOS Internet software and describes how to configure MPLS applications on the router.

This manual fully documents configurations for MPLS applications in Release 5.0 of the JUNOS Internet software.

To obtain additional information about the JUNOS software—either corrections to information in this manual or information that might have been omitted from this manual—refer to the software release notes.

To obtain the most current version of this manual and the most current version of the software release notes, refer to the product documentation page on the Juniper Networks Web site, which is located at <http://www.juniper.net/>.

To order printed copies of this manual or to order a documentation CD-ROM, which contains this manual, please contact your sales representative.

Audience

This manual is designed for network administrators who are configuring a Juniper Networks router. It assumes that you have a broad understanding of networks in general, the Internet in particular, networking principles, and network configuration. This manual assumes that you are familiar with one or more of the following Internet routing protocols: Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), Internet Control Message Protocol (ICMP), Resource Reservation Protocol (RSVP), Routing Information Protocol (RIP), and Simple Network Management Protocol (SNMP).

Document Organization

This manual is divided into several parts. Each part describes a major functional area of the JUNOS software, and the individual chapters within a part describe the software components of that functional area.

This manual contains the following parts and chapters:

- Part 1, “Overview,” provides an overview of Traffic Engineering concepts and configuration statements.
 - Chapter 1, “Traffic Engineering Overview,” provides a functional description of the four major components of traffic engineering, as well as the various methods of calculating and configuring label-switched paths (LSPs).
 - Chapter 2, “Complete MPLS Applications Configuration Mode Statements,” provides a comprehensive list of all the configuration statements available and displays their hierarchy.
- Part 2, “MPLS,” describes how to configure the JUNOS software to support Multiprotocol Label Switching.
 - Chapter 3, “MPLS Overview,” provides a short overview of Multiprotocol Label Switching (MPLS), which provides a mechanism for engineering network traffic patterns that is independent of the shortest path determined by a routing protocol.
 - Chapter 4, “MPLS Configuration Statements,” lists all minimum and optional configuration statements for MPLS.
 - Chapter 5, “Configure MPLS Signaled LSPs,” describes how to configure the ingress router for signaled LSPs and how to enable RSVP.
 - Chapter 6, “Configure Static LSPs,” describes how to configure the ingress, intermediate, and egress routers for static MPLS.
 - Chapter 7, “Configure Explicit-Path LSPs,” describes how to manually configure LSPs by specifying each router hop between the ingress and egress routers.
 - Chapter 8, “Configure Miscellaneous MPLS Properties,” describes how to configure MPLS to gather statistics and trace protocol packets and operations, as well as how to control syslog messages and SNMP traps.
 - Chapter 9, “Summary of MPLS Configuration Statements,” lists all statements used to configure MPLS.

- Part 3, “RSVP,” describes how to configure the JUNOS software to support Resource Reservation Protocol.
 - Chapter 10, “RSVP Overview,” provides a short overview of the Resource Reservation Protocol (RSVP), a setup protocol designed to interact with integrated services on the Internet to request a specific quality of service (QoS).
 - Chapter 11, “RSVP Configuration Guidelines,” describes the minimum and optional configurations for RSVP.
 - Chapter 12, “Summary of RSVP Configuration Statements,” describes all RSVP configuration statements.
- Part 4, “LDP,” describes how to configure the JUNOS software to support Label Distribution Protocol.
 - Chapter 13, “LDP Overview,” provides a short overview of the Label Distribution Protocol (LDP), a protocol used to establish LSPs through a network by mapping network-layer routing information directly to data-link label-switched paths.
 - Chapter 14, “Configure LDP,” describes the minimum and optional configurations for LDP.
 - Chapter 15, “Summary of LDP Configuration Statements,” describes all LDP configuration statements.
- Part 5, “CCC,” describes how to configure the JUNOS software to support circuit cross-connect (CCC).
 - Chapter 16, “CCC Overview,” describes the types of CCCs.
 - Chapter 17, “CCC Configuration,” describes how to configure CCC configuration statements.
 - Chapter 18, “Summary of CCC Configuration Statements,” describes all CCC configuration statements.
- Part 6, “VPNs,” describes how to configure the JUNOS software to support Virtual Private Networks (VPNs).
 - Chapter 19, “Layer 3 VPN Overview,” provides an overview of Layer 3 VPNs.
 - Chapter 20, “Layer 3 VPN Configuration Guidelines,” describes the minimum and optional configurations for Layer 3 VPNs.
 - Chapter 21, “Layer 3 VPN Configuration Troubleshooting Guidelines,” provides guidance for troubleshooting Layer 3 VPNs.
 - Chapter 22, “Layer 3 VPN Configuration Examples,” provides configuration examples for Layer 3 VPNs.
 - Chapter 23, “Summary of Layer 3 VPN Configuration Statements,” describes the statements used to configure Layer 3 VPNs.
 - Chapter 24, “Layer 2 VPN Overview,” provides an overview of Layer 2 VPNs.

- Chapter 25, “Layer 2 VPN Configuration Guidelines,” describes the minimum and optional configurations for Layer 2 VPNs.
- Chapter 26, “Layer 2 VPN Configuration Example,” provides configuration examples for Layer 2 VPNs.
- Chapter 27, “Summary of Layer 2 VPN Configuration Statements,” describes the statements used to configure Layer 2 VPNs.

This manual also contains a glossary, a complete index, and an index of statements and commands.

Related Documentation

The following additional documentation describes the JUNOS Internet software:

- *JUNOS Internet Software Configuration Guide: Getting Started*—Provides an overview of the JUNOS Internet software and describes how to install and upgrade the software. This manual also describes how to configure system management functions and how to configure the chassis, including user accounts, passwords, and redundancy.
- *JUNOS Internet Software Configuration Guide: Interfaces, Class of Service, and Firewalls*—Provides an overview of the interface, class-of-service, and firewall functions of the JUNOS Internet software and describes how to configure the interfaces on the router.
- *JUNOS Internet Software Configuration Guide: Multicast*—Provides an overview of multicast concepts and describes how to configure multicast routing protocols.
- *JUNOS Internet Software Configuration Guide: Network Management*—Provides an overview of network management concepts and describes how to configure various network management features, such as SNMP, accounting options, and cflowd.
- *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*—Provides an overview of routing concepts and describes how to configure routing, routing policy, and unicast routing protocols.
- *JUNOS Internet Software Operational Mode Command Reference*—Describes the JUNOS Internet software operational mode commands you use to monitor and troubleshoot Juniper Networks routers.
- *JUNOScript API Guide*—Describes how to use the JUNOScript API to monitor and configure Juniper Networks routers.
- *JUNOScript API Reference*—Provides a reference page for each tag in the JUNOScript API.

Manual Part Organization

The parts in this manual typically contain the following chapters:

- Overview—Provides background information about and discusses concepts related to the software component described in that part of the book.
- Configuration statements—Lists all the configuration statements available to configure the software component. These chapters provide an overview of the configuration statement hierarchy for each software component.
- Configuration guidelines—Describes how to configure all features of the software component. The first section of the configuration guidelines describes the minimum configuration for that component, listing the configuration statements you must include to enable the software component on the router with only the bare minimum functionality. The remaining sections in the configuration guidelines are generally arranged so that the most common features are near the beginning.
- Statement summary—A reference that lists all configuration statements alphabetically and explains each statement and all its options. The explanation of each configuration statement consists of the following parts:
 - Syntax—Describes the full syntax of the configuration statement. For an explanation of how to read the syntax statements, see “Documentation Conventions” on page xxiv.
 - Hierarchy level—Tells where in the configuration statement hierarchy you include the statement.
 - Description—Describes the function of the configuration statement.
 - Options—Describes the configuration statement’s options if there are any. For options with numeric values, the allowed range and default value, if any, are listed. For multiple options, if one option is the default, that fact is stated. If a configuration statement is at the top of a hierarchy of options that are other configuration statements, these options are generally explained separately in the statement summary section.
 - Usage guidelines—Points to the section or sections in the configuration guidelines section that describe how to use the configuration statement.
 - Required privilege level—Indicates the permissions that the user must have to view or modify the statement in the router configuration. For an explanation of the permissions, see the *JUNOS Internet Software Configuration Guide: Installation and System Management*.
 - See also—Indicates other configuration statements that might provide related or similar functionality.

Using the Indexes

This manual contains two indexes: a complete index, which contains all index entries, and an index that contains only statements and commands.

In the index, bold page numbers point to pages in the statement summary sections of configuration chapters. The index entry for each configuration statement always contains at least two entries. The first, with a bold page number on the same line as the statement name, references the statement summary section. The second entry, “usage guidelines,” references the section in the configuration guidelines section that describes how to use the statement.

Documentation Conventions

General Conventions

In general text, this manual uses the following conventions:

- Statements, commands, filenames, directory names, IP addresses, and configuration hierarchy levels are shown in a sans serif font. In the following example, “stub” is a statement name and “[edit protocols ospf area *area-id*]” is a configuration hierarchy level:

To configure a stub area, include the stub statement at the [edit protocols ospf area *area-id*] hierarchy level.

- In examples, text that you type literally is shown in a bold font. In the following example, you type the word “show”:

```
[edit protocols ospf area area-id]  
cli# show  
stub <default-metric metric>
```

- Examples of command output are generally shown in a fixed-width font to preserve the column alignment. For example:

```
> show interfaces terse  
Interface      Admin Link Proto Local              Remote  
at-1/3/0       up    up           inet  1.0.0.1             --> 1.0.0.2  
at-1/3/0.0     up    up           inet  1.0.0.1             --> 1.0.0.2  
                iso  
fxp0           up    up  
fxp0.0         up    up    inet  192.168.5.59/24
```


Conventions for Software Commands and Statements

When describing the JUNOS software, this manual uses the following type and presentation conventions:

- Statement or command names that you type literally are shown in a nonitalicized font. In the following example, the statement name is “area”:

You configure all these routers by including the following area statement at the [edit protocols ospf] hierarchy level:

- Options, which are variable terms for which you substitute appropriate values, are shown in italics. In the following example, “area-id” is an option. When you type the area statement, you substitute a value for *area-id*.

area *area-id*;

- Optional portions of a configuration statement are enclosed in angle brackets. In the following example, the “default-metric *metric*” portion of the statement is optional:

stub <default-metric *metric*>;

- For text strings separated by a pipe (|), you must specify either *string1* or *string2*, but you cannot specify both or neither of them. Parentheses are sometimes used to group the strings.

string1 | *string2*
(*string1* | *string2*)

In the following example, you must specify either broadcast or multicast, but you cannot specify both:

broadcast | multicast

- For some statements, you can specify a set of values. The set must be enclosed in square brackets. For example:

community *name* members [*community-id*]

- The configuration examples in this manual are generally formatted in the way that they appear when you issue a show command. This format includes braces ({ }) and semicolons. When you type configuration statements in the CLI, you do not type the braces and semicolons. However, when you type configuration statements in an ASCII file, you must include the braces and semicolons. For example:

```
[edit]
cli# set routing-options static route default next-hop address retain
[edit]
cli# show
routing-options {
  static {
    route default {
      next-hop address;
      retain;
    }
  }
}
```

- Comments in the configuration examples are shown either preceding the lines that the comments apply to, or more often, on the same line. When comments are shown on the same line, they are preceded by a pound sign (#) to indicate where the comment starts. In an actual configuration, comments can only precede a line; they cannot be on the same line as a configuration statement. For example:

```

protocols {
  mpls {
    interface (interface-name | all);    # Required to enable MPLS on the interface
  }
  rsvp {
    interface interface-name;          # Required for dynamic MPLS only
  }
}

```

- The general syntax descriptions provide no indication of the number of times you can specify a statement, option, or keyword. This information is provided in the text of the statement summary.

Documentation Feedback

We are always interested in hearing from our customers. Please let us know what you like and do not like about the Juniper Networks documentation, and let us know of any suggestions you have for improving the documentation. Also, let us know if you find any mistakes in the documentation. Send your feedback to tech-doc@juniper.net.

How to Request Support

For technical support, contact Juniper Networks at support@juniper.net, or at 888-314-JTAC within the United States and 408-745-2121 from outside the United States.

Part 1

Overview

- Traffic Engineering Overview on page 3
- Complete MPLS Applications Configuration Mode Statements on page 9

.....

Chapter 1

Traffic Engineering Overview

The task of mapping traffic flows onto an existing physical topology is called *traffic engineering*. Traffic engineering provides the ability to move traffic flow away from the shortest path selected by the Interior Gateway Protocol (IGP) and onto a potentially less congested physical path across a network.

Traffic engineering provides the capabilities to:

- Route primary paths around known bottlenecks or points of congestion in the network.
- Provide precise control over how traffic is rerouted when the primary path is faced with single or multiple failures.
- Provide more efficient use of available aggregate bandwidth and long-haul fiber by ensuring that subsets of the network do not become overutilized while other subsets of the network along potential alternate paths are underutilized.
- Maximize operational efficiency.
- Enhance the traffic-oriented performance characteristics of the network by minimizing packet loss, minimizing prolonged periods of congestion, and maximizing throughput.
- Enhance statistically bound performance characteristics of the network (such as loss ratio, delay variation, and transfer delay) required to support a multiservices Internet.

This chapter discusses the following topics:

- Components of Traffic Engineering on page 3
- Flexible LSP Calculation and Configuration on page 6

Components of Traffic Engineering

In the JUNOS software, traffic engineering is implemented with Multiprotocol Label Switching (MPLS) and the Resource Reservation Protocol (RSVP). Traffic engineering is composed of four functional components:

- Packet Forwarding Component on page 4
- Information Distribution Component on page 5

- Path Selection Component on page 5

- Signaling Component on page 6

Packet Forwarding Component

The packet forwarding component of the JUNOS traffic engineering architecture is MPLS, which is responsible for directing a flow of IP packets along a predetermined path across a network. This path is called a *label-switched path (LSP)*. LSPs are simplex; that is, the traffic flows in one direction from the head-end (ingress) router to a tail-end (egress) router. Duplex traffic requires two LSPs: one LSP to carry traffic in each direction. An LSP is created by the concatenation of one or more label-switched hops, allowing a packet to be forwarded from one router to another across the MPLS domain.

When an ingress router receives an IP packet, it adds an MPLS header to the packet and forwards it to the next router in the LSP. The labeled packet is forwarded along the LSP by each router until it reaches the tail end of the LSP, at which point the MPLS header is removed and the packet is forwarded based on Layer 3 information such as the IP destination address. The key purpose in this scheme is that the physical path of the LSP not be limited to what the IGP would choose as the shortest path to reach the destination IP address.

Packet Forwarding Based on Label Swapping

The packet forwarding process at each router is based on the concept of *label swapping*. This concept is similar to what occurs at each ATM switch in a PVC. Each MPLS packet carries a 4-byte encapsulation header that contains a 20-bit, fixed-length label field. When a packet containing a label arrives at a router, the router examines the label and uses it as an index into its MPLS forwarding table. Each entry in the forwarding table contains an interface-inbound label pair mapped to a set of forwarding information that is applied to all packets arriving on the specific interface with the same inbound label.

Example of How a Packet Traverses an MPLS Backbone

This section describes how an IP packet is processed as it traverses an MPLS backbone network.

At the entry edge of the MPLS backbone, the IP header is examined by the ingress router. Based on this analysis, the packet is classified, assigned a label, encapsulated in an MPLS header, and forwarded toward the next hop in the LSP. MPLS provides a high degree of flexibility in the way that an IP packet can be assigned to an LSP. For example, in the JUNOS traffic engineering implementation, all packets arriving at the ingress router that are destined to exit the MPLS domain at the same egress router are forwarded along the same LSP.

Once the packet begins to traverse the LSP, each router uses the label to make the forwarding decision. The MPLS forwarding decision is made independent of the original IP header: the incoming interface and label are used as lookup keys into the MPLS forwarding table. The old label is replaced with a new label, and the packet is forwarded to the next hop along the LSP. This process is repeated at each router in the LSP until the packet reaches the egress router.

When the packet arrives at the egress router, the label is removed and the packet exits the MPLS domain. The packet is then forwarded based on the destination IP address contained in the packet's original IP header according to the traditional shortest path calculated by the IP routing protocol.

Information Distribution Component

Traffic engineering requires detailed knowledge about the network topology as well as dynamic information about network loading. The information distribution component is implemented by defining relatively simple extensions to the IGP's so that link attributes are included as part of each router's link-state advertisement. IS-IS extensions include the definition of new Type Length Values (TLVs), while OSPF extensions are implemented with opaque link-state advertisements (LSAs). The standard flooding algorithm used by the link-state IGP's ensures that link attributes are distributed to all routers in the routing domain. Some of the traffic engineering extensions to be added to the IGP link-state advertisement include maximum link bandwidth, maximum reserved link bandwidth, current bandwidth reservation, and link coloring.

Each router maintains network link attributes and topology information in a specialized traffic engineering database (TED). The TED is used exclusively for calculating explicit paths for the placement of LSPs across the physical topology. A separate database is maintained so that the subsequent traffic engineering computation is independent of the IGP and the IGP's link-state database. Meanwhile, the IGP continues its operation without modification, performing the traditional shortest-path calculation based on information contained in the router's link-state database.

Path Selection Component

After network link attributes and topology information are flooded by the IGP and placed in the TED, each ingress router uses the TED to calculate the paths for its own set of LSPs across the routing domain. The path for each LSP can be represented by either a strict or loose explicit route. An explicit route is a preconfigured sequence of routers that should be part of the physical path of the LSP. If the ingress router specifies all the routers in the LSP, the LSP is said to be identified by a *strict* explicit route. If the ingress router specifies only some of the routers in the LSP, the LSP is described as a *loose* explicit route. Support for strict and loose explicit routes allows the path selection process to be given broad latitude whenever possible, but to be constrained when necessary.

The ingress router determines the physical path for each LSP by applying a Constrained Shortest Path First (CSPF) algorithm to the information in the TED. CSPF is a shortest-path-first algorithm that has been modified to take into account specific restrictions when calculating the shortest path across the network. Input into the CSPF algorithm includes:

- Topology link-state information learned from the IGP and maintained in the TED
- Attributes associated with the state of network resources (such as total link bandwidth, reserved link bandwidth, available link bandwidth, and link color) that are carried by IGP extensions and stored in the TED
- Administrative attributes required to support traffic traversing the proposed LSP (such as bandwidth requirements, maximum hop count, and administrative policy requirements) that are obtained from user configuration

As CSPF considers each candidate node and link for a new LSP, it either accepts or rejects a specific path component based on resource availability or whether selecting the component violates user policy constraints. The output of the CSPF calculation is an explicit route consisting of a sequence of router addresses that provides the shortest path through the network that meets the constraints. This explicit route is then passed to the signaling component, which establishes forwarding state in the routers along the LSP.

Offline Planning and Analysis

Despite the reduced management effort resulting from online path calculation, an offline planning and analysis tool is still required to optimize traffic engineering globally. Online calculation takes resource constraints into account and calculates one LSP at a time. The challenge with this approach is that it is not deterministic. The order in which an LSP is calculated plays a critical role in determining its physical path across the network. LSPs that are calculated early in the process have more resources available to them than LSPs calculated later in the process because previously calculated LSPs consume network resources. If the order in which the LSPs are calculated is changed, the resulting set of physical paths for the LSPs also can change.

An offline planning and analysis tool simultaneously examines each link's resource constraints and the requirements of each LSP. While the offline approach can take several hours to complete, it performs global calculations, compares the results of each calculation, and then selects the best solution for the network as a whole. The output of the offline calculation is a set of LSPs that optimizes utilization of network resources. After the offline calculation is completed, the LSPs can be established in any order because each is installed following the rules for the globally optimized solution.

Signaling Component

An LSP is not known to be workable until it is actually established by the signaling component. The signaling component, which is responsible for establishing LSP state and distributing labels, relies on a number of extensions to the Resource Reservation Protocol (RSVP):

- The Explicit Route Object allows an RSVP PATH message to traverse an explicit sequence of routers that is independent of conventional shortest-path IP routing. Recall that the explicit route can be either strict or loose.
- The Label Request Object permits the RSVP PATH message to request that intermediate routers provide a label binding for the LSP that it is establishing.
- The Label Object allows RSVP to support the distribution of labels without having to change its existing mechanisms. Because the RSVP RESV message follows the reverse path of the RSVP PATH message, the Label Object supports the distribution of labels from downstream nodes to upstream nodes.

Flexible LSP Calculation and Configuration

Traffic engineering involves mapping traffic flow onto a physical topology. You can determine the paths online using constraint-based routing. Regardless of how the physical path is calculated, the forwarding state is installed across the network using RSVP.

The JUNOS software supports a number of different ways to route and configure an LSP:

- You can calculate the full path for the LSP offline and individually configure each router in the LSP with the necessary static forwarding state. This is analogous to how some ISPs currently configure their IP-over-ATM cores.
- You can calculate the full path for the LSP offline and statically configure the ingress router with the full path. The ingress router then uses RSVP as a dynamic signaling protocol to install a forwarding state in each router along the LSP.

- You can rely on constraint-based routing to perform dynamic online LSP calculation. You configure the constraints for each LSP, and then the network itself determines the path that best meets those constraints. Specifically, the ingress router calculates the entire LSP based on the constraints and then initiates signaling across the network.
- You can calculate a partial path for an LSP offline and statically configure the ingress router with a subset of the routers in the path and then permit online calculation to determine the complete path.

For example, consider a topology that includes two east-west paths across the United States: one in the north through Chicago and one in the south through Dallas. If you want to establish an LSP between a router in New York and one in San Francisco, you can configure the partial path for the LSP to include a single loose-routed hop of a router in Dallas. The result is an LSP routed along the southern path. The ingress router uses CSPF to compute the complete path and uses RSVP to install the forwarding state along the LSP.

- You can configure the ingress router with no constraints whatsoever. In this case, normal IGP shortest-path routing is used to determine the path of the LSP. This configuration does not provide any value in terms of traffic engineering. However, it is easy and might be useful in situations when services such as Virtual Private Networks (VPNs) are needed.

In all these cases, you can specify any number of LSPs as backups for the primary LSP, thus allowing you to combine more than one configuration approach. For example, you might explicitly compute the primary path offline, set the secondary path to be constraint-based, and have the tertiary path be unconstrained. If a circuit on which the primary LSP is routed fails, the ingress router notices the outage from error notifications received from a downstream router or by the expiration of RSVP soft-state information. Then, the router dynamically forwards traffic to a hot-standby LSP or calls on RSVP to create a forwarding state for a new backup LSP.

Chapter 2

Complete MPLS Applications Configuration Mode Statements

This chapter shows the complete configuration statement hierarchy for the MPLS applications configuration statements, listing all possible configuration statements and showing their level in the configuration hierarchy. When you are configuring the JUNOS software, your current hierarchy level is shown in the banner on the line preceding the `user@host#` prompt.

For a complete list of the JUNOS configuration statements, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

This chapter is organized as follows:

- [edit protocols connections] Hierarchy Level on page 10
- [edit protocols l2vpn] Hierarchy Level on page 10
- [edit protocols ldp] Hierarchy Level on page 11
- [edit protocols mpls] Hierarchy Level on page 11
- [edit protocols rsvp] Hierarchy Level on page 13
- [edit routing-instances] Hierarchy Level (for Layer 2 VPNs) on page 14
- [edit routing-instances] Hierarchy Level (for Layer 3 VPNs) on page 14

[edit protocols connections] Hierarchy Level

```

protocols {
  connections {
    interface-switch connection-name {
      interface interface-name.unit-number;
      interface interface-name.unit-number;
    }
    lsp-switch connection-name {
      transmit-lsp label-switched-path;
      receive-lsp label-switched-path;
    }
    remote-interface-switch connection-name {
      interface interface-name.unit-number;
      transmit-lsp label-switched-path;
      receive-lsp label-switched-path;
    }
  }
}

```

[edit protocols l2vpn] Hierarchy Level

```

protocols {
  l2vpn {
    encapsulation-type <type>
    traceoptions {
      file filename <replace> <size size> <files number> <nostamp>;
      flag flag <flag-modifier> <disable>;
    }
    site site-name {
      site-identifier identifier;
      interface interface-name {
        site-offset offset;
      }
    }
  }
}

```

[edit protocols ldp] Hierarchy Level

```

protocols {
  ldp {
    import [policy-name];
    deaggregate | no-deaggregate;
    egress-policy policy-name;
    export [policy-name];
    keepalive-interval seconds;
    keepalive-timeout seconds;
    preference preference;
    transport-address ( interface | loopback );
    interface interface-name {
      disable;
      hello-interval seconds;
      hold-time seconds;
      deaggregate | no-deaggregate;
      transport-address ( interface | loopback );
    }
    traceoptions {
      file filename <replace> <size size> <files number> <no-stamp>
        <(world-readable | no-world-readable)>;
      flag flag <flag-modifier> <disable>;
    }
  }
}

```

[edit protocols mpls] Hierarchy Level

```

protocols {
  mpls {
    disable;
    admin-groups {
      group-name group-value;
    }
    log-updown {
      (syslog | no-syslog);
      (trap | no-trap);
    }
    no-propagate-ttl;
    optimize-aggressive;
    path path-name {
      address <strict | loose>;
    }
    statistics {
      file filename size size files number <no-stamp>;
      interval seconds;
    }
    traceoptions {
      file filename <replace> <size size> <files number> <no-stamp>
        <(world-readable | no-world-readable)>;
      flag flag <flag-modifier> <disable>;
    }
  }
}

```

```

traffic-engineering (bgp | bgp-igp);
label-switched-path lsp-path-name {
  disable;
  to address;
  from address;
  adaptive;
  admin-group {
    exclude group-names;
    include group-names;
  }
  bandwidth bps;
  class-of-service cos-value;
  fast-reroute {
    bandwidth bps;
    exclude group-names;
    hop-limit number;
    include group-names;
  }
  hop-limit number;
  ldp-tunneling;
  metric number;
  no-cspf;
  no-decrement-ttl;
  optimize-timer seconds;
  preference preference;
  priority setup-priority hold-priority;
  (random | least-fill | most-fill);
  (record | no-record);
  retry-limit number;
  retry-timer seconds;
  standby;
  primary path-name {
    adaptive;
    admin-group {
      exclude group-names;
      include group-names;
    }
    bandwidth bps;
    class-of-service cos-value;
    hop-limit number;
    no-cspf;
    optimize-timer seconds;
    preference preference;
    priority setup-priority hold-priority;
    (record | no-record);
    standby;
  }
  secondary path-name {
    adaptive;
    admin-group {
      exclude group-names;
      include group-names;
    }
    bandwidth bps;
    class-of-service cos-value;
    hop-limit number;
    no-cspf;
    optimize-timer seconds;
    preference preference;
    priority setup-priority hold-priority;
    (record | no-record);
    standby;
  }
}

```

```

install {
    destination-prefix/prefix-length <active>;
}
}
interface (interface-name | all) {
    disable;
    admin-group {
        group-name;
    }
    label-map in-label {
        (nexthop (address | interface-name | address/interface-name)) | (reject | discard);
        (pop | (swap <out-label>));
        class-of-service value;
        preference preference;
        type type;
    }
}
static-path inet {
    prefix {
        nexthop (address | interface-name | address/interface-name);
        push out-label;
        class-of-service value;
        preference preference;
    }
}
}
}

```

[edit protocols rsvp] Hierarchy Level

```

protocols {
    rsvp {
        disable;
        keep-multiplier number;
        preemption ( aggressive | disabled | normal );
        refresh-time seconds;
        traceoptions {
            file filename <replace> <size size> <files number> <no-stamp>
            <(world-readable | no-world-readable)>;
            flag flag <flag-modifier> <disable>;
        }
        interface interface-name {
            disable;
            (aggregate | no-aggregate);
            authentication-key key;
            bandwidth bps;
            hello-interval seconds;
            subscription percentage;
        }
    }
}
}

```

[edit routing-instances] Hierarchy Level (for Layer 2 VPNs)

For Layer 2 VPNs, only some of the statements in the [edit routing-instances] hierarchy are valid. For the full hierarchy, see *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

```
[edit]
routing-instances {
  routing-instance-name {
    instance-type l2vpn;
    interface interface-name;
    route-distinguisher ( as-number:id | ip-address:id );
    protocols {
      l2vpn {
        l2vpn configuration
      }
    }
  }
}
```

[edit routing-instances] Hierarchy Level (for Layer 3 VPNs)

For Layer 3 VPNs, only some of the statements in the [edit routing-instances] hierarchy are valid. For the full hierarchy, see *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

```
[edit]
routing-instances {
  routing-instance-name {
    interface interface-name;
    instance-type (forwarding | no-forwarding | vrf);
    route-distinguisher ( as-number:number | ip-address:number );
    vrf-import [ policy-name ];
    vrf-export [ policy-name ];
    protocols {
      bgp {
        bgp-configuration;
      }
      ospf {
        ospf-configuration;
      }
      rip {
        rip-configuration;
      }
    }
    routing-options {
      autonomous-system autonomous-system <loops number>;
      forwarding-table {
        export [ policy-name ];
      }
      interface-routes {
        rib-group group-name;
      }
      martians {
        destination-prefix match-type <allow>;
      }
      options {
        syslog (level level | upto level);
      }
      rib routing-table {
```



```

static {
  defaults {
    static-options;
  }
  route destination-prefix {
    next-hop;
    static-options;
  }
}
martians {
  destination-prefix match-type <allow>;
}
static {
  defaults {
    static-options;
  }
  route destination-prefix {
    policy [ policy-name ];
    static-options;
  }
}
}
router-id address;
static {
  defaults {
    static-options;
  }
  route destination-prefix {
    policy [ policy-name ];
    static-options;
  }
}
}
}

```

.....

Part 2

MPLS

- MPLS Overview on page 19
- MPLS Configuration Statements on page 39
- Configure MPLS Signaled LSPs on page 43
- Configure Static LSPs on page 73
- Configure Explicit-Path LSPs on page 79
- Configure Miscellaneous MPLS Properties on page 81
- Summary of MPLS Configuration Statements on page 85

Chapter 3

MPLS Overview

Multiprotocol Label Switching (MPLS) provides a mechanism for engineering network traffic patterns that is independent of routing tables. MPLS assigns short labels to network packets that describe how to forward them through the network. MPLS is independent of any routing protocol and can be used for unicast packets.

In the traditional Level 3 forwarding paradigm, as a packet travels from one router to the next, an independent forwarding decision is made at each hop. The IP network layer header is analyzed, and the next hop is chosen based on this analysis and on the information in the routing table. In an MPLS environment, the analysis of the packet header is performed just once, when a packet enters the MPLS cloud. The packet then is assigned to a stream, which is identified by a *label*, which is a short (20-bit), fixed-length value at the front of the packet. Labels are used as lookup indexes into the label forwarding table. For each label, this table stores forwarding information. You can associate additional information with a label—such as class-of-service (CoS) values—that can be used to prioritize packet forwarding.

This chapter discusses the following topics:

- MPLS Standards on page 19
- Link-Layer Support on page 20
- MPLS and Traffic Engineering on page 20
- MPLS Applications on page 33
- MPLS and Routing Tables on page 36
- MPLS and Traffic Protection on page 38

MPLS Standards

The JUNOS software supports the following RFCs and Internet drafts related to MPLS:

- *ICMP Extensions for Multiprotocol Label Switching*, Internet draft
draft-ietf-mpls-icmp-02.txt

The following documents provide a good overview of MPLS:

- RFC 3031, *Multiprotocol Label Switching Architecture*
- RFC 3032, *MPLS Label Stack Encoding*

The following documents provide information about traffic engineering:

- RFC 2702, *Requirements for Traffic Engineering Over MPLS*
- *IS-IS Extensions for Traffic Engineering*, Internet draft draft-ietf-isis-traffic-02.txt
- *Traffic Engineering Extensions to OSPF*, Internet draft draft-katz-yeung-ospf-traffic-04.txt

To access Internet RFCs and drafts, go to the IETF web site at <http://www.ietf.org>.

The JUNOS software supports a proprietary MIB for MPLS objects; see the *JUNOS Internet Software Configuration Guide: Interfaces and Chassis* for more information.

Link-Layer Support

MPLS supports the following link-layer protocols, which are all supported in the JUNOS MPLS implementation:

- PPP—Protocol ID 0x0281, NCP protocol ID 0x8281.
- Ethernet/Cisco HDLC—Ethernet type 0x8847.
- ATM—SNAP-encoded Ethernet type 0x8847. Support is included for both point-to-point mode or NBMA mode. Support is not included for encoding MPLS labels as part of ATM VPI/VCI.
- Frame Relay—SNAP-encoded, Ethernet type 0x8847. Support is not included for encoding MPLS labels as part of Frame Relay DLCI.
- GRE Tunnel—Ethernet type 0x8847.

MPLS and Traffic Engineering

Traffic engineering allows you to control the path that data packets follow, bypassing the standard routing model, which uses routing tables. Traffic engineering moves flows from congested links to alternate links that would not be selected by the automatically computed destination-based shortest path. With traffic engineering, you can:

- Make more efficient use of expensive long-haul fibers.
- Control how traffic is rerouted in the face of single or multiple failures.
- Classify critical and regular traffic on a per-path basis.

The core of the traffic engineering design is based on building label-switched paths (LSPs) among routers. An LSP is connection-oriented, like a virtual circuit in Frame Relay or ATM. LSPs are not reliable: packets entering an LSP do not have delivery guarantees, although preferential treatment is possible. LSPs also are similar to unidirectional tunnels in that packets entering a path are encapsulated in an envelope and switched across the entire path without being touched by intermediate nodes. LSPs provide fine-grained control over how packets are forwarded in a network. To provide reliability, an LSP can use a set of primary and secondary paths.

LSPs can be configured for only BGP traffic (traffic whose destination is outside of an AS). In this case, traffic within the AS is not affected by the presence of LSPs. LSPs can also be configured for both BGP and IGP traffic; therefore, both intra-AS and inter-AS traffic is affected by the LSPs.

Label Description

Packets travelling along an LSP are identified by a *label*, a 20-bit, unsigned integer in the range 0 through 1048575:

- 0 through 15—Reserved and have special semantics.
- 16 through 1023—Unused and unassigned by the software, a feature that is specific to the JUNOS software. You can use labels to manually configure static LSPs and to ensure that there are no conflicts with labels that are dynamically assigned by the software.
- 1024 through 99,999—Reserved for future applications.
- 100,000 through 1,048,575—Automatically negotiated, assigned, released, and reused by the software. Typically, per-box labels are assigned in the 100,000-799,999 range, and per-interface labels are assigned in the 800,000-1,048,575 range.

Special Labels

Some of the reserved labels (in the 0 through 15 range) have well-defined meanings. For more complete details, see RFC 3032, *MPLS Label Stack Encoding*.

- 0, IPv4 Explicit Null Label—This value is legal only when it is the sole label entry (no label stacking). It indicates that the label must be popped upon receipt. Forwarding continues based on the IPv4 packet.
- 1, Router Alert Label—When a packet is received with a top label value of 1, it is delivered to the local software module for processing.
- 2, IPv6 Implicit Null Label—This value is legal only when it is the sole label entry (no label stacking). It indicates that the label must be popped upon receipt. Forwarding continues based on the IPv6 packet.
- 3, Implicit Null Label—This label is used in the control protocol (LDP, RSVP) only to request label popping by the downstream router. It never actually appears in the encapsulation. Labels with a value of 3 should not be used in the data packet as real labels. No payload type (IPv4 or IPv6) is implied with this label.
- 4 through 15—Unassigned.

Special labels are commonly used between the egress and penultimate routers of an LSP. If the LSP is configured to carry IPv4 packets only, the egress router might signal the penultimate router to use 0 as a final hop label. If the LSP is configured to carry IPv6 packets only, the egress router might signal the penultimate router to use 2 as a final hop label.

The egress router might simply signal the penultimate router to use 3 as the final label, which is a request to perform penultimate hop label popping. This means an egress router will not process a labelled packet; rather, it receives the payload (IPv4, IPv6, or others) directly. This reduces one MPLS lookup at egress.

For label-stacked packets, the egress router receives an MPLS label packet with its top label already popped by the penultimate router. The egress router cannot receive label-stacked packets using label 0 or 2.

When functioning as an egress router, JUNOS software Release 4.2 and later typically requests label 3. In JUNOS software Release 4.0 and earlier, the egress router typically requests label 0 from the penultimate router. There are no interoperability problems among various JUNOS software versions because the software accepts any of these special labels and performs the requested operations.

Label Allocation

Note that earlier versions of JUNOS software allocate labels on a per-interface basis. Labels on different interfaces are assigned independently. This means that a particular label received on one interface is not related to the same label received on a different interface. For this reason, labels usually are preceded by an interface name in display output (in the format *interface.label*). For example, so-5/0/0.0.01024 indicates that the label value 01024 was received on interface so-5/0/0.0.

In the JUNOS software Release 4.2 and later, label values are allocated per router only. The display output shows only the label (for example, 01024).

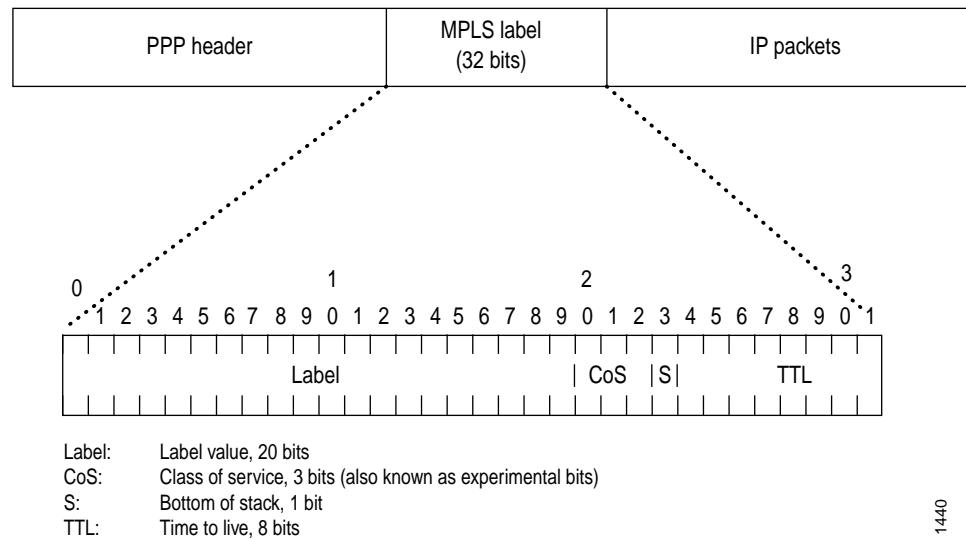
Labels for multicast packets are independent of those for unicast packets. Currently, the JUNOS software does not support multicast labels.

Labels are assigned by downstream routers relative to the flow of packets. A router receiving labeled packets (the next-hop router) is responsible for assigning incoming labels. A received packet containing a label that is unrecognized (unassigned) is dropped. For unrecognized labels, the router does not attempt to unwrap the label to analyze the network layer header, nor does it generate an ICMP destination unreachable message.

A packet can carry a number of labels, organized as a last-in, first-out stack. This is referred to as a *label stack*. At a particular router, the decision as to how to forward a labeled packet is based exclusively on the label at the top of the stack.

Figure 1 shows the encoding of a single label. The encoding appears after data link layer headers, but before any network layer header.

Figure 1: Label Encoding



1440

Operations on Labels

The router supports the following label operations:

- **Push**—Add a new label to the top of the packet. For IPv4 packets, the new label is the first label. The TTL, S, and CoS fields are derived from the IP packet header. If the Push operation is performed on an existing MPLS packet, you will have a packet with 2 or more labels. This is called label stacking. The top label must have its S field set to 0, and might derive CoS and TTL from lower levels. Note that in JUNOS software Release 4.2 and later, the new top label in a label stack always initializes its TTL to 255, regardless of the TTL value of lower labels.
- **Pop**—Remove the label from the beginning of the packet. Once the label is removed, the TTL is copied from the label into the IP packet header, and the underlying IP packet is forwarded as a native IP packet. In the case of multiple labels in a packet (label stacking), removal of the top label yields another MPLS packet. The new top label might derive CoS and TTL from a previous top label. Note that in JUNOS software Release 4.2 and later, the popped TTL value from the previous top label is not written back to the new top label.
- **Swap**—Replaces the label at the top of the label stack with a new label. The S and CoS bits are copied from the previous label, and the TTL value is copied and decremented (unless the no-decrement-ttl or no-propagate-ttl statements are configured). A transit router supports a label stack of any depth.
- **Multiple Push**—Add multiple labels (up to 3) on top of existing packets. This is equivalent to doing Push multiple times.
- **Swap and Push**—Replace the existing top of the label stack with a new label, followed by pushing another new label on top.

Routers in an LSP

Each router in an LSP performs one of the following functions:

- Ingress router—The router at the beginning of an LSP. This router encapsulates IP packets with an MPLS Layer 2 frame and forwards it to the next router in the path. Each LSP can have only one ingress router.
- Egress router—The router at the end of an LSP. This router removes the MPLS encapsulation, thus transforming it from an MPLS packet to an IP packet, and forwards the packet to its final destination using information in the IP forwarding table. Each LSP can have only one egress router. The ingress and egress routers in an LSP cannot be the same router.
- Transit router—Any intermediate router in the LSP between the ingress and egress routers. A transit router forwards received MPLS packets to the next router in the MPLS path. An LSP can contain zero or more transit routers, up to a maximum of 253 transit routers in a single LSP.

A single router can be part of multiple LSPs. It can be the ingress or egress router for one or more LSPs, and it also can be a transit router in one or more LSPs. The functions that each router supports depend on your network design.

How a Packet Travels along an LSP

When an IP packet enters an LSP, the ingress router examines the packet and assigns it a label based on its destination, placing the label in the packet's header. The label transforms the packet from one that is forwarded based on its IP routing information to one that is forwarded based on information associated with the label.

The packet then is forwarded to the next router in the LSP. This router and all subsequent routers in the LSP do not examine any of the IP routing information in the labeled packet. Rather, they use the label to look up information in their label forwarding table. Then, they replace the old label with a new label and forward the packet to the next router in the path.

When the packet reaches the egress router, the label is removed, and the packet again becomes a native IP packet and is again forwarded based on its IP routing information.

Types of LSPs

There are three types of LSPs:

- Static LSPs—For static paths, you must manually assign labels on all routers involved (ingress, transit, and egress). No signaling protocol is needed. This procedure is similar to configuring static routes on individual routers. Like static routes, there is no error reporting, no liveness detection, and no statistics reporting.
- LDP signaled LSPs—See LDP Overview on page 139.

- **RSVP signaled LSPs**—For signaled paths, RSVP is used to set up the path and dynamically assign labels. (RSVP signaling messages are used to set up signaled paths.) You configure only the ingress router. The transit and egress routers accept signaling information from the ingress router, and they set up and maintain the LSP cooperatively. Any errors encountered while establishing an LSP are reported to the ingress router for diagnostics. For signaled LSPs to work, a version of RSVP that supports tunnel extensions must be enabled on all routers.

There are two types of RSVP signaled LSPs:

- **Explicit-path LSPs**—All intermediate hops of the LSP are manually configured. The intermediate hops can be strict, loose, or any combination of strict and loose hops. Explicit path LSPs provide you with complete control over how the path is set up. They are similar to static LSPs but require much less configuration.
- **Constrained-path LSPs**—The intermediate hops of the LSP are automatically computed by the software. The computation takes into account information provided by the topology information from the IS-IS or OSPF link-state routing protocol, the current network resource utilization determined by RSVP, and the resource requirements and constraints of the LSP. For signaled constrained-path LSPs to work, either the IS-IS or OSPF protocol and the IS-IS or OSPF traffic engineering extensions must be enabled on all routers.

Scope of LSPs

For LSPs that use constrained-path, the LSP computation is confined to one IGP area, and cannot cross any AS boundary. This prevents an AS from extending its IGP into another AS.

Explicit-path LSPs, however, can cross as many AS boundaries as necessary. Because intermediate hops are manually specified, the LSP has no dependence on the IGP topology or a local forwarding table.

Constrained-Path LSP Computation

The Constrained Shortest Path First (CSPF) algorithm is an advanced form of the Shortest Path First (SPF) algorithm used in OSPF and IS-IS route computations. CSPF is used in computing paths for LSPs that are subject to multiple constraints. When computing paths for LSPs, CSPF considers not only the topology of the network, but also the attributes of the LSP and the links, and it attempts to minimize congestion by intelligently balancing the network load.

The constraints that CSPF considers include:

- **LSP attributes**
 - Bandwidth requirements
 - Hop limitations
 - Administrative groups (that is, link color requirements)
 - Priority (setup and hold)
 - Explicit route (strict or loose)

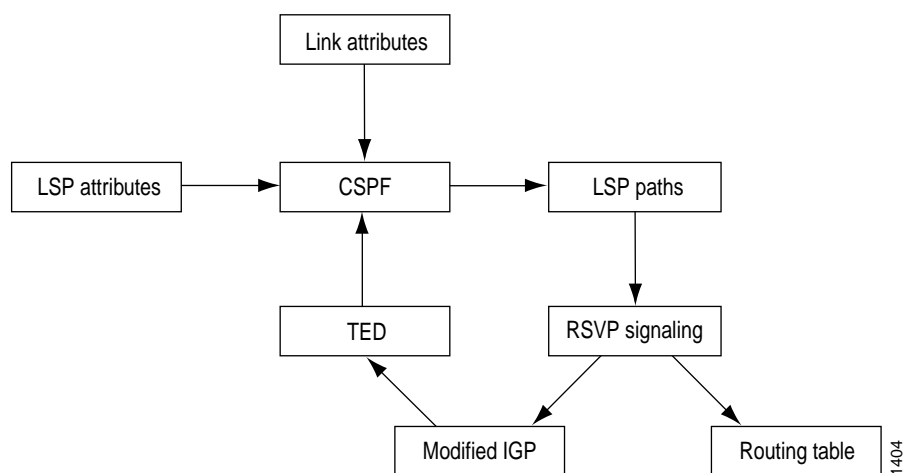
- Link attributes

- Reservable bandwidth of the links (static bandwidth minus the currently reserved bandwidth)
- Administrative groups (that is, link colors assigned to the link)

The data that CSPF considers comes from the:

- Traffic Engineering Database (TED)—Provides CSPF with up-to-date topology information, the current reservable bandwidth of links, and the link colors. For the CSPF algorithm to perform its computations, a link-state IGP (such as OSPF or IS-IS) with special extensions is needed. For CSPF to be effective, the link-state IGP on all routers must support the special extensions. While building the topology database, the extended IGP must take into consideration the current LSPs and must flood the route information everywhere. Because changes in the reserved link bandwidth and link color cause database updates, an extended IGP tends to flood more frequently than a normal IGP. See Figure 2 for a diagram of the relationships between these components.
- Currently active LSPs—Includes all the LSPs that should originate from the router and their current operational status (up, down, or timeout).

Figure 2: CSPF Computation Process



How CSPF Selects a Path

To select a path, CSPF follows these steps:

1. Compute LSPs one at a time, beginning with the highest priority LSP (the one with the lowest setup priority value). Among LSPs of equal priority, CSPF starts with those that have the highest bandwidth requirement.
2. Prune the topology database (TED) of all the links that are not full duplex and do not have sufficient reservable bandwidth.
3. If the LSP configuration includes the include statement, prune all links that do not share any included colors.

4. If the LSP configuration includes the exclude statement, prune all links that contain excluded colors and do not contain a color.
5. Find the shortest path towards the LSP's egress router, taking into account explicit-path constraints. For example, if the path must pass through Router A, two separate SPF's are computed, one from the ingress router to Router A, the other from Router A to the egress router.
6. If several paths have equal cost, choose the one whose last hop address is the same as the LSP's destination.
7. If several equal-cost paths remain, select the one with the fewest number of hops.
8. If several equal-cost paths remain, apply the CSPF load-balancing rule configured on the LSP (least-fill, most-fill, or random).

Path Selection Tie-Breaking

If more than one path is available after applying the rules from the previous section, a tie-breaking rule is applied to choose the path for the LSP. There are three tie-breaking rules: random, least fill, and most fill. The rule used depends on the configuration. Random is the default rule. For other rules, the following definitions are needed:

reservable bandwidth = bandwidth of link x subscription factor of link

available bandwidth = reservable bandwidth - (sum of the bandwidths of the LSPs traversing the link)

available bandwidth ratio = available bandwidth/reservable bandwidth

minimum available bandwidth ratio (for a path) = the smallest available bandwidth ratio of the links in a path

- **Random**—One of the remaining paths is picked at random. This rule tends to place an equal number of LSPs on each link, regardless of the available bandwidth ratio.
- **Least fill**—The path with the largest minimum available bandwidth ratio is preferred. This rule tries to equalize the reservation on each link.
- **Most fill**—The path with the smallest minimum available bandwidth ratio is preferred. This rule tries to fill a link before moving on to alternative links.

Computing Paths Offline

The JUNOS software provides online, real-time CSPF computation only; each router performs CSPF calculations independent of the other routers in the network. These calculations are based on currently available topology information—information that is usually recent, but not completely accurate. LSP placements are locally optimized, based on current network status.

To optimize links globally across the network, you can use an offline tool to perform the CSPF calculations and determine the paths for the LSPs. You can create such a tool yourself, or you can modify an existing network design tool to perform these calculations. You should run the tool periodically (daily or weekly) and download the results into the router. An offline tool should take the following into account when performing the optimized calculations:

- All the LSP's requirements
- All link attributes
- Complete network topology

Fate Sharing

Fate sharing allows you to create a database of information that CSPF uses to compute one or more backup paths to use in case the primary path becomes unstable. The database describes the relationships between elements of the network, such as routers and links. You can specify one or more elements within a group.

Through fate sharing, you can configure backup paths that minimize the number of shared links and fiber paths with the primary paths as much as possible, to ensure that if a fiber is cut, the minimum amount of data is lost and that a path still exists to the destination.

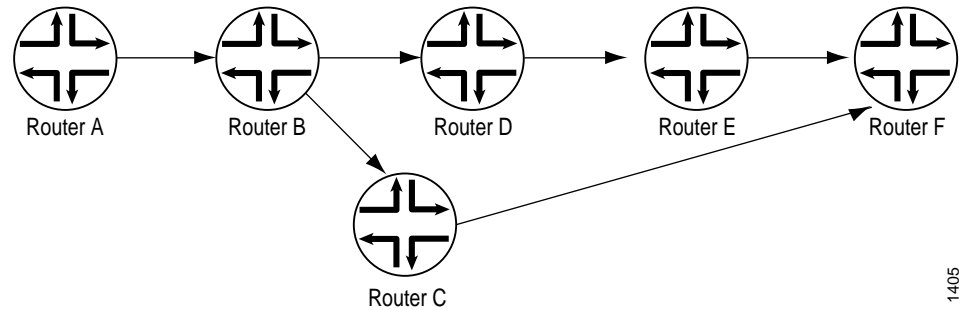
For a backup path to work optimally, it must not share links or physical fiber paths with the primary path. This ensures that a single point of failure will not affect the primary and backup paths at the same time. For more information on fate sharing, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

IGP Shortcuts

Link-state protocols, such as OSPF and IS-IS, use the SPF algorithm to compute the shortest-path tree to all nodes in the network. The results of such computations can be represented by the destination node, next-hop address, and output interface, where the output interface is a physical interface. LSPs can be used to augment the SPF algorithm. On the node performing the calculations, LSPs appear to be logical interfaces directly connected to remote nodes in the network. If you configure the IGP to treat LSPs the same as a physical interface and to use the LSPs as a potential output interface, the SPF computation results are represented by the destination node and output LSP, effectively using the LSP as a shortcut through the network to the destination.

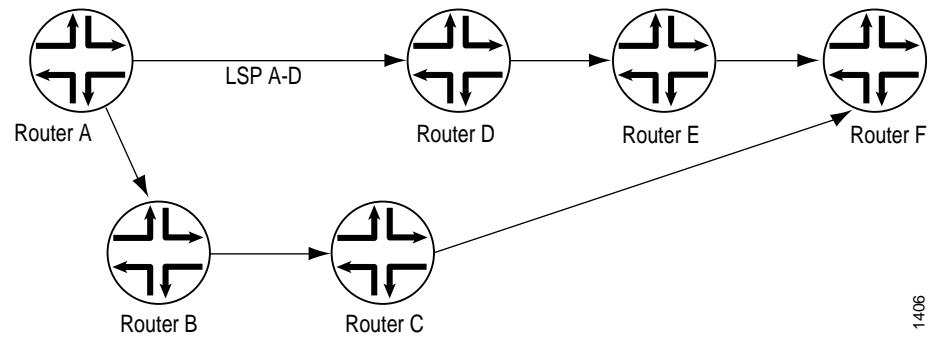
As an illustration, begin with a typical SPF tree (Figure 3):

Figure 3: Typical SPF Tree, Sourced from Router A



If an LSP connects Router A to Router D and if IGP shortcuts are enabled on Router A, you might have the SPF tree shown in Figure 4.

Figure 4: Modified SPF Tree, Using LSP A-D as a Shortcut



Note that Router D is now reachable through LSP A-D. When computing the shortest path to reach Router D, Router A has two choices:

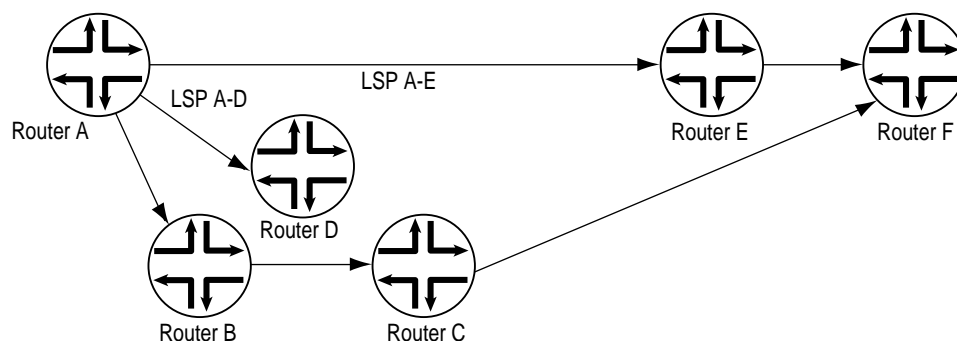
- Use IGP path A-B-D.
- Use LSP A-D.

Router A decides between the two choices by comparing the IGP metrics for path A-B-D with the LSP metrics for LSP A-D. If the IGP metric is lower, path A-B-D is chosen (Figure 3). If the LSP metric is lower, LSP A-D is used (Figure 4). If both metrics are equal, Router A might share the load between the two paths.

Note that Routers E and F are also reachable through LSP A-D, because they are downstream from Router D in the SPF tree.

Assuming another LSP connects Router A to Router E, you might have the SPF tree shown in Figure 5.

Figure 5: Modified SPF Tree, Using Both LSP A-D and LSP A-E as Shortcuts



Enable IGP Shortcuts

IGP shortcuts are supported for both IS-IS and OSPF. A link-state protocol is required for IGP shortcuts. Shortcuts are disabled by default. For information about enabling IGP shortcuts for IS-IS and OSPF, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*. You can enable IGP shortcuts on a per-router basis; you do not need to enable shortcuts globally. A router's shortcut computation does not depend on another router performing similar computations, and shortcuts performed by other routers are irrelevant.

LSPs Qualified in Shortcut Computations

Not all LSPs are used in IGP shortcuts. Only those LSPs whose egress point (using the `to` statement) matches the router ID of the egress node are considered. Other LSPs, whose egress point matches the egress node interface address, are ignored in IGP shortcuts.

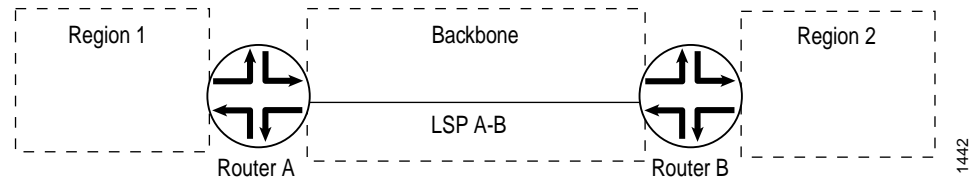
There are exceptions, however. If an LSP has an alias egress point (using the `install` statement), and it matches certain router IDs, it is included in the shortcut computation as well. If multiple equal metric LSPs destined to the same router ID exist, traffic can load-share among them.

IGP Shortcut Applications

You can use shortcuts to engineer traffic traveling towards destination nodes that do not support MPLS LSPs. For example, in Figure 5, traffic traveling toward Router F enters LSP A-E. You can control traffic between Router A and Router F by manipulating LSP A-E; you do not need to explicitly set up an LSP between Router A and Router F.

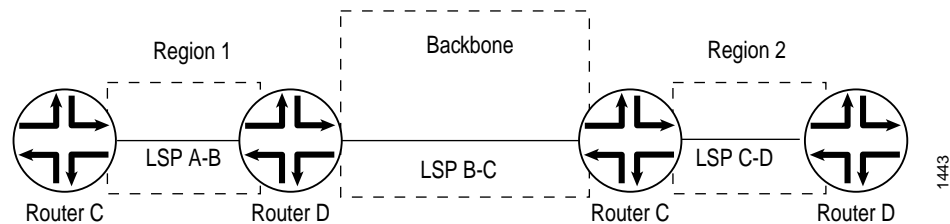
In Figure 6, all traffic from Region 1 to Region 2 traverses LSP A-B if IGP shortcuts are enabled on the ingress router (Router A), permitting aggregation of interregional traffic into one LSP. To perform traffic engineering on the interregional traffic, you only have to manipulate LSP A-B, which avoids creating N^2 LSPs from all routers in Region 1 to all routers in Region 2 and allows efficient resource controls on the backbone network.

Figure 6: IGP Shortcuts



Shortcuts allow you to deploy LSPs into a network in an incremental, hierarchical fashion. In Figure 7, each region can choose to implement traffic engineering LSPs independently, without requiring cooperation from other regions. Each region can choose to deploy intraregion LSPs to fit the region's bandwidth needs, at the pace appropriate for the region.

Figure 7: IGP Shortcuts in a Bigger Network



When intraregion LSPs are in place, interregional traffic automatically traverses the intraregion LSPs as needed, eliminating the need for a full mesh of LSPs between edge routers. For example, traffic from Router A to Router D traverses LSPs A-B, B-C, and C-D.

IGP Shortcuts and Routing Table

IGP typically does two independent computations. The first one is performed without considering any LSP. The result of the computation is stored in the inet.0 table. This step is no different from traditional SPF computations and is always performed even if IGP shortcut is disabled.

The second computation is performed with only LSPs as a logical interface in mind, producing routes that are reachable through LSPs only. The results are stored in the inet.3 table only. The routes produced in the second step are typically a subset of the first step.

If traffic engineering for IGP and BGP is enabled (see "IGP and BGP Destinations" on page 35), IGP moves all routes in inet.3 into inet.0, merging all routes, and at the same time emptying the inet.3 table. The number of routes in inet.0 will be exactly the same as before. Route next hops may traverse a physical interface, an LSP, or the combination of both if the metrics are equal.

Router Requirements

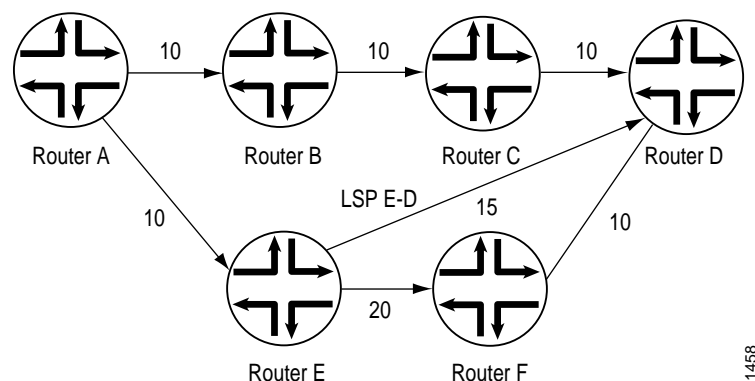
IGP shortcuts are enabled on a per-node basis. You do not need to coordinate with other nodes.

Advertise LSPs into IGP

IGP shortcuts allow an ingress router of an LSP to use the LSP in its SPF computation. However, other routers on the network do not know of the existence of that LSP, so they cannot use it. This can lead to suboptimal traffic engineering.

As an example, consider the network shown in Figure 8:

Figure 8: SPF Computations with Advertised LSPs



Assume that Router A is computing a path to Router D. The link between Router E and Router F has metric 20; all other links have metric 10. Here, the path chosen by Router A is A-B-C-D, which has a metric of 30, instead of A-E-F-D, which has a metric of 40.

If Router E has an LSP to Router D with a metric of 15, you want traffic from Router A to Router D to use the path A-E-D, which has a metric of 25, instead of the path A-B-C-D. However, because Router A does not know about the LSP between Router E and Router D, it cannot route traffic through this path.

For all routers on the network to know about the LSP between Router E and Router D, you need to advertise it. This advertisement announces the LSP as a unidirectional, point-to-point link in the link-state database, and all routers can compute paths using the LSP. The link-state database maintains information about the Autonomous System topology and contains information about the router's local state (for example, the router's usable interfaces and reachable neighbors). In Figure 3, Router A will see the link from Router E to Router D and route traffic along this lower-metric path.

Because an LSP is announced as a unidirectional link, you might need to configure a reverse LSP (one that starts at the egress router and ends at the ingress router) so that the SPF bidirectionality check succeeds. As a step in the SPF computation, IS-IS considers a link from Router E to Router D. Before IS-IS uses any link, it verifies that there is a link from Router D to Router E (there is bidirectional connectivity between router E and D). Otherwise, the SPF computation will not use an announced LSP.

MPLS Applications

In the JUNOS implementation of MPLS, establishing an LSP installs on the ingress router a host route (a 32-bit mask) toward the egress router. The address of the host route is the destination address of the LSP. By default, the route has a preference value of 7, a value that is higher than all routes except direct interface and static routes. The 32-bit mask ensures the route is more specific (that is, longer match) than all other subnet routes. The host routes can be used to traffic-engineer BGP destinations only, or both IGP and BGP destinations.

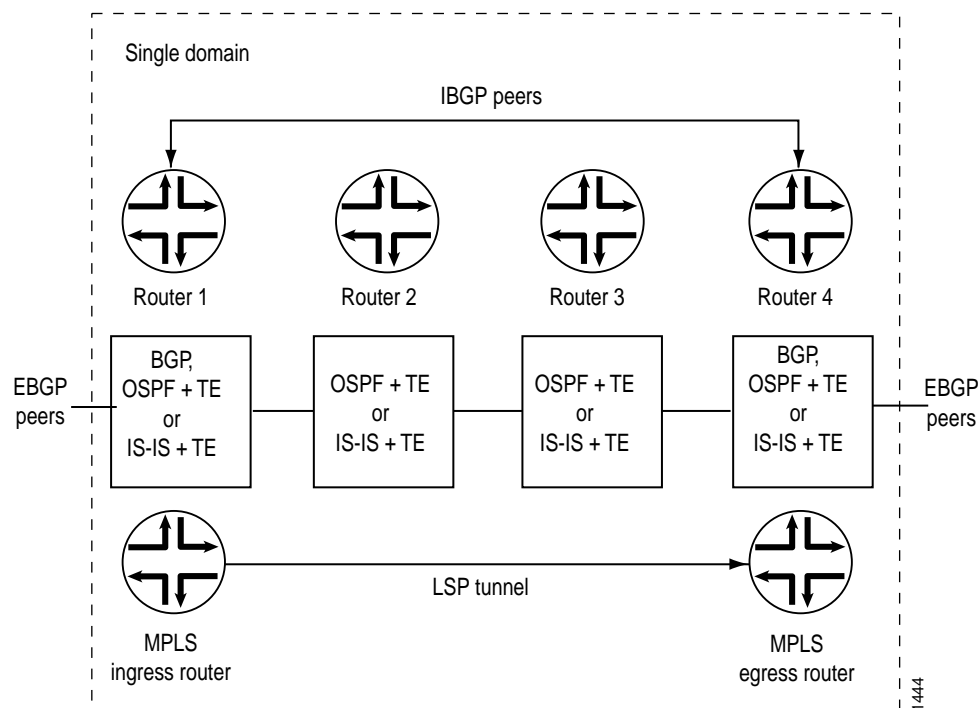
BGP Destinations

You can configure MPLS to control the paths that traffic takes to destinations outside an AS.

Both IBGP and EBGp take advantage of the LSP host routes without requiring extra configuration. BGP compares the BGP next-hop address with the LSP host route. If a match is found, the packets for the BGP route are label-switched over the LSP. If multiple BGP routes share the same next-hop address, all the BGP routes are mapped to the same LSP route, regardless of which BGP peer the routes are learned from. If the BGP next-hop address does not match an LSP host route, BGP routes continue to be forwarded based on the IGP routes within the routing domain. In general, when both an LSP route and an IGP route exist for the same BGP next-hop address, the one with the highest preference is chosen.

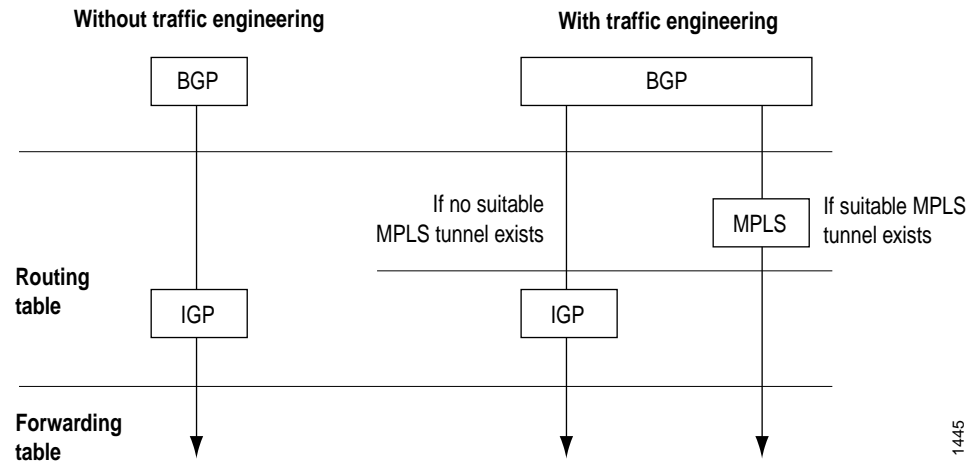
Figure 9 shows an MPLS topology that illustrates how MPLS and LSPs work. This topology consists of a single domain with four routers. The two routers at the edges of the domain, Router 1 and Router 4, are running EBGp to communicate with peers outside the domain and IBGP to communicate between themselves. For intradomain communication, all four routers are running an IGP. Finally, an LSP tunnel exists from Router 1 to Router 4.

Figure 9: MPLS Application Topology



When BGP on Router 1 receives prefixes from Router 4, it must determine how to reach a BGP next-hop address. Typically, when traffic engineering is not enabled, BGP uses IGP routes to determine how to reach next-hop addresses. (See the left side of Figure 10.) However, when traffic engineering is enabled, if the BGP next hop matches the LSP tunnel end point (that is, the MPLS egress router), those prefixes enter the LSP tunnel. (To track these prefixes, look at the Active Route field in the `show mpls lsp` command output or at the output of the `show route label-switched-path path-name` command.) If the BGP next hop does not match an LSP tunnel end point, those prefixes are sent following the IGP's shortest path. (See the right side of Figure 10.)

Figure 10: How BGP Determines How to Reach Next-Hop Addresses



1445

IGP and BGP Destinations

You can configure MPLS to control the paths that traffic takes to destinations within an AS.

When traffic engineering is for IGP destinations only, the MPLS host routes are installed in the inet.3 routing table (see Figure 11), separate from the routes learned from other routing protocols. Not all inet.3 routes are downloaded into the forwarding table. Packets directly addressed to the egress router do not follow the LSP, which prevents routes learned from LSPs from overriding routes learned from IGP or other sources.

Traffic within a domain, including BGP control traffic between BGP peers, is not affected by LSPs. MPLS affects interdomain transit traffic only; that is, it affects only those BGP prefixes that are learned from an external domain. MPLS does not disrupt intradomain traffic, so IS-IS or OSPF routes remain undisturbed. If you issue a ping or traceroute command to any destination within the domain, the ping or traceroute packets follow the IGP path. However, if you issue a ping or traceroute command from Router 1 in Figure 9 (the LSP ingress router) to a destination outside of the domain, the packets use the LSP tunnel.

When traffic engineering for IGP and BGP destinations is enabled, the MPLS host routes are installed in the inet.0 table (see Figure 12) and downloaded into the forwarding table. Any traffic destined to the egress router could enter the LSP. In effect, it moves all the routes in inet.3 into inet.0, causing the inet.3 table to be emptied.

RSVP packets automatically avoid all MPLS LSPs, including those established by RSVP or LDP. This prevents placing one RSVP session into another LSP, or in other words, nesting one LSP into another.

Select Forwarding LSP Next Hop

If more than one LSP tunnel to a BGP next hop exists, the prefixes learned from the BGP next hop are randomly divided among the LSP tunnels. To control which LSP BGP uses to forward data for a given prefix, use the install-nexthop statement in the export policy applied to the forwarding table. For more information, see *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

MPLS and Routing Tables

The IGP and BGP store their routing information in the routing table `inet.0`, which is the main IP routing table. If `traffic-engineering bgp` is configured, thereby allowing only BGP to use MPLS paths for forwarding traffic, MPLS path information is stored in a separate routing table, `inet.3`. Only BGP accesses the `inet.3` routing table. BGP uses both `inet.0` and `inet.3` to resolve next-hop addresses. If `traffic-engineering bgp-igp` is configured, thereby allowing the IGP to use MPLS paths for forwarding traffic, MPLS path information is stored in the `inet.0` routing table. (Figure 11 and Figure 12 illustrate the routing tables in the two traffic-engineering configurations.)

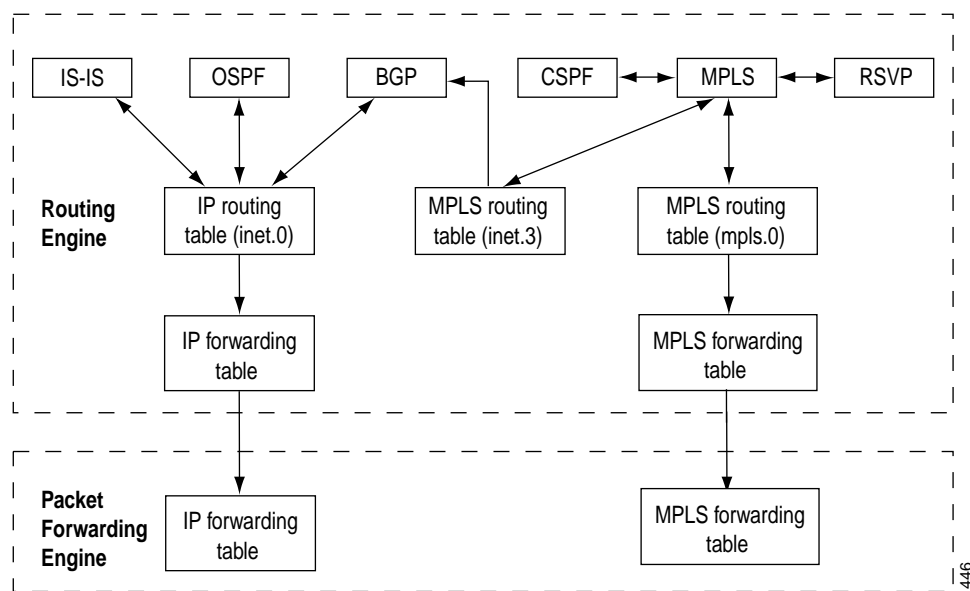
The `inet.3` routing table contains the host address of each LSP's egress router. This routing table is used on ingress routers to route packets to the destination egress router. BGP uses the `inet.3` routing table on the ingress router to help in resolving next-hop addresses.

MPLS also maintains an MPLS path routing table (`mpls.0`), which contains a list of the next label-switched router in each LSP. This routing table is used on transit routers to route packets to the next router along an LSP.

Typically, the egress router in an LSP does not consult the `mpls.0` routing table. (This router does not need to consult `mpls.0` because the penultimate router in the LSP either changes the packet's label to a value of 0 or pops the label.) In either case, the egress router forwards it as an IPv4 packet, consulting the IP routing table, `inet.0`, to determine how to forward the packet.

When a transit or egress router receives an MPLS packet, information in the MPLS forwarding table is used to determine the next transit router in the LSP or to determine that this router is the egress router.

Figure 11: MPLS Routing and Forwarding Tables When `traffic-engineering bgp` Is Configured



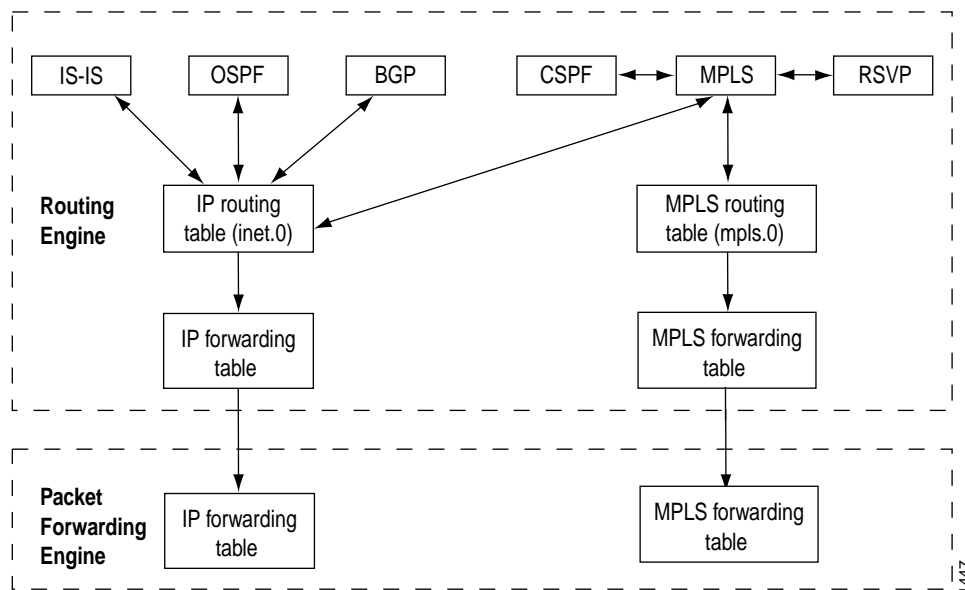
1446

When BGP resolves a next-hop prefix, it examines both the inet.0 and inet.3 routing tables, seeking the next hop with the highest preference. If it finds a next-hop entry with equal preference in both routing tables, BGP prefers the entry found in the inet.3 routing table.

Generally, BGP selects next-hop entries in the inet.3 routing table, because their preferences are always lower than OSPF and IS-IS next-hop preferences. When you configure LSPs, you can override the default preference for MPLS LSPs, which might alter the next-hop selection process.

When BGP selects a next-hop entry from the inet.3 routing table, it installs that LSP into the forwarding table in the Packet Forwarding Engine, which causes packets destined for that next hop to enter and travel along the LSP. If the LSP is removed or fails, the path is removed from the inet.3 routing table and from the forwarding table, and BGP reverts to using a next hop from the inet.0 routing table.

Figure 12: MPLS Routing and Forwarding Tables When traffic-engineering bgp-igp Is Configured



MPLS and Traffic Protection

Typically, when an LSP fails, the router immediately upstream from the failure signals the outage to the ingress router. The ingress router calculates a new path to the egress router, establishes the new LSP, and then directs the traffic from the failed path to the new path. This rerouting process can be time-consuming and prone to failure. For example, the outage signals to the ingress router might get lost, or the new path might take too long to come up, resulting in significant packet drops. The JUNOS software provides two complementary mechanisms for protecting against LSP failures:

- **Standby secondary paths**—You can configure primary and secondary paths. You configure secondary paths with the `standby` statement. To activate traffic protection, you need to configure these standby paths only on the ingress router. If the primary path fails, the ingress router immediately reroutes traffic from the failed path to the standby path, thereby eliminating the need to calculate a new route and signal a new path. For more information about configuring standby LSPs, see “Configure the Standby State” on page 64.
- **Fast reroute**—You configure fast reroute on an LSP to minimize the effect of a failure in the LSP. Fast reroute enables a router upstream from the failure to route around the failure quickly to the router downstream of the failure. The upstream router then signals the outage to the ingress router, thereby maintaining connectivity before a new LSP is established. For more information about fast reroute, see “Configure Fast Reroute” on page 49.

When standby secondary path and fast reroute are both configured on the LSP, full traffic protection is enabled. When a failure occurs in an LSP, the router upstream of the failure routes traffic around the failure and notifies the ingress router of the failure. This rerouting keeps the traffic flowing while waiting for the notification to be processed at the ingress router. After receiving the failure notification, the ingress router immediately reroutes the traffic from the patched primary path to the more optimal standby path.

Per-Prefix Load Balancing

When there are multiple equal cost tunnels to a destination, load balancing can be controlled for each path. Load balancing is proportional to the configured bandwidth per LSP. If an LSP has a larger bandwidth associated with it, that LSP will carry a larger number of prefixes. If you configure the bandwidth, the prefixes automatically adjust themselves.

Chapter 4

MPLS Configuration Statements

To configure MPLS, you can include the following statements in the configuration:

```
protocols {
  mpls {
    disable;
    admin-groups {
      group-name group-value;
    }
    advertise-hold-time seconds;
    log-updown {
      (syslog | no-syslog);
      (trap | no-trap);
    }
    no-propagate-ttl;
    optimize-aggressive;
    path path-name {
      address <strict | loose>;
    }
    statistics {
      file filename size size files number <no-stamp>;
      interval seconds;
    }
    traceoptions {
      file filename <replace> <size size> <files number> <no-stamp>
        <(world-readable | no-world-readable)>;
      flag flag <flag-modifier> <disable>;
    }
    traffic-engineering (bgp | bgp-igp);
    label-switched-path lsp-path-name {
      disable;
      to address;
      from address;
      adaptive;
      admin-group {
        exclude group-names;
        include group-names;
      }
      bandwidth bps;
      class-of-service cos-value;
      fast-reroute {
        bandwidth bps;
        (exclude group-names | no-exclude);
        hop-limit number;
        (include group-names | no-include);
      }
    }
  }
}
```

```

hop-limit number;
ldp-tunneling;
metric number;
no-cspf;
no-decrement-ttl;
optimize-timer seconds;
preference preference;
priority setup-priority hold-priority;
(random | least-fill | most-fill);
(record | no-record);
retry-limit number;
retry-timer seconds;
standby;
primary path-name {
    adaptive;
    admin-group {
        exclude group-names;
        include group-names;
    }
    bandwidth bps;
    class-of-service cos-value;
    hop-limit number;
    no-cspf;
    optimize-timer seconds;
    preference preference;
    priority setup-priority hold-priority;
    (record | no-record);
    standby;
}
secondary path-name {
    adaptive;
    admin-group {
        exclude group-names;
        include group-names;
    }
    bandwidth bps;
    class-of-service cos-value;
    hop-limit number;
    no-cspf;
    optimize-timer seconds;
    preference preference;
    priority setup-priority hold-priority;
    (record | no-record);
    standby;
}
install {
    destination-prefix/prefix-length <active>;
}
}
interface (interface-name | all) {
    disable;
    admin-group {
        group-name;
    }
    label-map in-label {
        (nexthop (address | interface-name | address/interface-name)) | (reject | discard);
        (pop | (swap <out-label>));
        class-of-service value;
        preference preference;
        type type;
    }
}
}

```

```

static-path inet {
    prefix {
        nexthop (address | interface-name | address/interface-name);
        push out-label;
        class-of-service value;
        preference preference;
    }
}
}
}

```

Minimum MPLS Configuration

To enable MPLS on the router, you must include at least the following statements. All other MPLS configuration statements are optional. Note that this configuration does nothing more than enable MPLS on the router and on the specified interface.

```

[edit]
interfaces {
    interface-name {
        logical-unit-number {
            family mpls;
        }
    }
}
protocols {
    mpls {
        interface (interface-name | all);
    }
    rsvp {
        interface interface-name;
    }
}

```

For every interface you enable, two special routes are installed automatically in the MPLS forwarding table. One route has a label value of 0, and the second has a label value of 1. (For information on these labels, see “Special Labels” on page 21.)

Chapter 5

Configure MPLS Signaled LSPs

To configure MPLS signaled LSPs, you create an LSP that runs from the ingress router to the egress router. (For information on LDP signaled LSPs, see “Configure LDP” on page 145.) To create the LSP, you configure only the ingress router; you do not have to configure any other routers. You can configure the LSP so that the JUNOS software makes all forwarding decisions, or you can configure some or all routers in the path. The LSP is set up by RSVP, through RSVP signaling messages. The JUNOS software automatically negotiates, assigns, releases, and reuses labels. Automatically assigned labels have a value from 1024 through 1048575.

To configure signaled LSPs across a network, perform the following tasks:

- Configure the Ingress Router for Signaled LSPs on page 43
- Configure All Other MPLS Routers for Signaled LSPs on page 68
- Enable RSVP on page 68
- Configure MPLS over GRE Tunnels on page 71

For a configuration example, see “Examples: Configure Signaled LSPs” on page 68.

Configure the Ingress Router for Signaled LSPs

To configure signaled LSPs, perform the following tasks on the ingress router:

- Create a Named Path on page 44
- Create an LSP on page 45
- Configure Alternate Backup Paths Using Fate Sharing on page 66

Create a Named Path

To configure signaled LSPs, you must first create one or more named paths on the ingress router. For each path, you can specify some or all transit routers in the path, or you can leave it empty.

Each path name can be up to 16 characters and can contain letters, digits, periods, and hyphens. The name must be unique within the ingress router. Once a named path is created, you can configure LSPs using the named path on the primary or on the secondary statement at the [edit protocols mpls label-switched-path *label-path-name*] hierarchy level. You can specify the same named path on any number of LSPs.

To create an empty path, create a named path by including the following form of the path statement at the [edit protocols mpls] hierarchy level. This form of the path statement is empty, which means that any path between the ingress and egress routers is accepted. In actuality, the path used tends to be the same path as is followed by destination-based, best-effort traffic.

```
[edit protocols mpls]
path path-name;
```

To create a path in which you specify some or all transit routers in the path, include the following form of the path statement at the [edit protocols mpls] hierarchy level, specifying one *address* for each transit router:

```
[edit protocols mpls]
path path-name {
    address | host name <strict | loose>;
}
```

In this form of the path statement, you specify one or more transit router addresses. Specifying the ingress and/or egress routers is optional. You can specify the address or host name of each transit router, although you do not need to list each transit router if its type is loose. Specify the addresses in order, starting with the ingress router (optional) or the first transit router, and continuing sequentially along the path up to the egress router (optional) or the router immediately before the egress router. You need to specify only one address per router hop. If you specify more than one address for the same router, only the first address is used; the additional addresses are ignored and truncated.

For each router address, you specify the type, which can be one of the following:

- **strict**—(Default) The route taken from the previous router to this router is a direct path and cannot include any other routers. If *address* is an interface address, this router also ensures that the incoming interface is the one specified. Doing this is useful when there are parallel links between the previous router and this router. It also ensures that routing can be enforced on a per-link basis.

For strict addresses, you must ensure that the router immediately preceding the router you are configuring has a direct connection to that router. The address can be a loopback interface address, in which case the incoming interface is not checked.

- **loose**—The route taken from the previous router to this router need not be a direct path and can include other routers and can be received on any interface. The address can be any interface address or the address of the loopback interface.

Examples: Create a Named Path

The following path, `to-hastings`, specifies the complete strict path from the ingress to the egress routers through 14.1.1.1, 13.1.1.1, 12.1.1.1 and 11.1.1.1, in that order. There cannot be any intermediate routers except the ones specified. However, there can be intermediate routers between 11.1.1.1 and the egress router because the egress router is not specifically listed in the path statement. To prevent intermediate routers before egress, configure the egress router as the last router, with a strict type.

```
[edit protocols mpls]
path to-hastings {
  14.1.1.1 strict;
  13.1.1.1 strict;
  12.1.1.1 strict;
  11.1.1.1 strict;
}
```

The following path, `alt-hastings`, allows any number of intermediate routers between routers 14.1.1.1 and 11.1.1.1. In addition, intermediate routers are permitted between 11.1.1.1 and the egress router.

```
[edit protocols mpls]
path alt-hastings {
  14.1.1.1 strict;
  11.1.1.1 loose;
}
```

Create an LSP

The second step in configuring signaled LSPs is to create one or more LSPs and define the properties associated with the label-switched path on the ingress router. To configure an LSP, include the `label-switched-path` statement at the `[edit protocols mpls]` hierarchy level:

```
[edit protocols mpls]
label-switched-path lsp-path-name {
  disable;
  to address;
  from address;
  adaptive;
  admin-group {
    exclude group-names;
    include group-names;
  }
  bandwidth bps;
  class-of-service cos-value;
  fast-reroute {
    fast-reroute bps;
    exclude group-names;
    hop-limit number;
    include group-names;
  }
  hop-limit number;
  ldp-tunneling;
  metric number;
  no-cspf;
  no-decrement-ttl;
  optimize-timer seconds;
  preference preference;
  priority setup-priority hold-priority;
}
```

```

    (random | least-fill | most-fill);
    (record | no-record);
    retry-limit number;
    retry-timer seconds;
    standby;
    primary path-name {
        adaptive;
        admin-group {
            exclude group-names;
            include group-names;
        }
        bandwidth bps;
        class-of-service cos-value;
        hop-limit number;
        no-cspf;
        optimize-timer seconds;
        preference preference;
        priority setup-priority hold-priority;
        (record | no-record);
        standby;
    }
    secondary path-name {
        adaptive;
        admin-group {
            exclude group-names;
            include group-names;
        }
        bandwidth bps;
        class-of-service cos-value;
        hop-limit number;
        no-cspf;
        optimize-timer seconds;
        preference preference;
        priority setup-priority hold-priority;
        (record | no-record);
        standby;
    }
}

```

Each LSP must have a name, *lsp-path-name*, which can be up to 32 characters long and can contain letters, digits, periods (.), and hyphens (-). The name must be unique within the ingress router. For ease of management and identification, configure unique names across the entire domain.

When you configure LSPs, you can specify the following statements either for each LSP or for each path. (You configure LSPs at the [edit protocols mpls label-switched-path *lsp-path-name*] hierarchy level, and you configure paths at the [edit protocols mpls label-switched-path *lsp-path-name* primary] or [edit protocols mpls label-switched-path *lsp-path-name* secondary] hierarchy level.) For statements that you configure on a per-LSP basis, the value applies to all paths in the LSP. For statements that you configure on a per-path basis, the path value overrides the per-LSP value.

- adaptive
- admin-group
- bandwidth
- class-of-service
- hop-limit

- no-cspf
- optimize-timer
- preference
- priority
- record or no-record
- standby

For each LSP, you can configure the following properties:

- Configure the Address of the Egress Router on page 48
- Configure the Address of the Ingress Router on page 48
- Configure the Primary and Secondary LSPs on page 49
- Configure Fast Reroute on page 49
- Configure Addresses to Associate with the LSP on page 52
- Configure Path Connection Retry Information on page 53
- Configure the Dynamic LSP Metric on page 53
- Configure the Static LSP Metric on page 54
- Configure CSPF Tie Breaking on page 54
- Configure Load-Balancing LSPs without CSPF on page 55
- Disable Normal TTL Decrementing on page 55

For each LSP and for each primary and secondary path, you can configure the following properties:

- Disable Constrained Path LSP Computation on page 56
- Configure Administrative Groups on page 57
- Configure the LSP Preference on page 59
- Configure Whether to Record Path Routes on page 59
- Configure the MPLS CoS Value on page 60
- Configure an LSP to Be Adaptive on page 61
- Configure Priority and Preemption on page 62
- Optimize Signaled LSPs on page 63
- Configure the Maximum Path Length on page 64

- Configure the Path Bandwidth on page 64
- Configure the Standby State on page 64
- Configure LSP Hold Time on page 65
- Configure LDP Tunneling on page 65

Configure the Address of the Egress Router

When configuring an LSP, you must specify the address of the egress router by including the `to` statement at the `[edit protocols mpls label-switched-path lsp-path-name]` hierarchy level:

```
[edit protocols mpls label-switched-path lsp-path-name]
  to address;
```

When you are setting up an LSP, the `to` statement is the only required statement. All other statements are optional.

After the LSP is established, the address of the egress router is installed as a host route in the routing table. Then, this route can be used by BGP to forward traffic.

To have the software send BGP traffic over an LSP, the address of the egress router is the same as the address of the BGP next hop. You can specify the egress router's address as any one of the router's interface addresses or as the BGP router ID. If you specify a different address, even if the address is on the same router, BGP traffic is not sent over the LSP.

To determine the address of the BGP next hop, use the `show route detail` command. To determine the destination address of an LSP, use the `show mpls lsp` command. To determine whether a route has gone through an LSP, use the `show route` or `show route forwarding-table` command. In the output of these last two commands, the `label-switched-path` or `push` keyword included with the route indicates it has passed through an LSP. Also, use the `traceroute` command to trace the actual path that the route leads to. This is another indication as to whether a route has passed through an LSP.

You also can manipulate the address of the BGP next hop by defining a BGP import policy filter that sets the route's next-hop address.

Configure the Address of the Ingress Router

The local router always is considered to be the ingress router, which is the beginning of the LSP. The software automatically determines the proper outgoing interface and IP address to use to reach the next router in an LSP.

By default, the router ID is chosen as the address of the ingress router. To override the automatic selection of the source address, specify a source address in the `from` statement at the `[edit protocols mpls label-switched-path lsp-path-name]` hierarchy level:

```
[edit protocols mpls label-switched-path lsp-path-name]
  from address;
```

The outgoing interface used by the LSP is not affected by the source address that you configure.

Configure the Primary and Secondary LSPs

By default, an LSP routes itself hop by hop toward the egress router. The LSP tends to follow the shortest path as dictated by the local routing table, usually taking the same path as destination-based, best-effort traffic. These paths are “soft” in nature because they automatically reroute themselves whenever a change occurs in a routing table or in the status of a node or link.

To configure the path so that it follows a particular route, create a named path using the path statement, as described in “Create a Named Path” on page 44. Then you apply the named path by including the primary or secondary statement at the [edit protocols mpls label-switched-path *lsp-path-name*] hierarchy level:

```
[edit protocols mpls label-switched-path lsp-path-name]
primary path-name {
  ...
}
secondary path-name {
  ...
}
```

A named path can be referenced by any number of LSPs.

The primary statement creates the primary path, which is the LSP’s preferred path. The secondary statement creates an alternative path. If the primary path can no longer reach the egress router, the alternative path is used.

When the software switches from the primary to a secondary path, it continuously attempts to revert to the primary path, switching back to it when it is again reachable, but no sooner than the retry time specified in the retry-timer statement. (For more information, see “Configure Path Connection Retry Information” on page 53.)

You can configure zero or one primary path. If you do not configure a primary path, the first secondary path that is established is selected as the path.

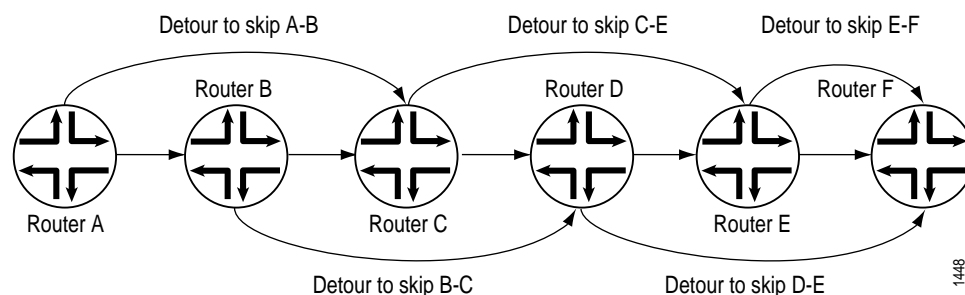
You can configure zero or more secondary paths. All secondary paths are equal, and the software tries them in the order that they are listed in the configuration. The software does not attempt to switch among secondary paths. If the current secondary path is not available, the next one is tried. To create a set of equal paths, specify secondary paths without specifying a primary path.

If you do not specify any named paths, or if the path that you specify is empty, the software makes all routing decisions necessary to reach the egress router.

Configure Fast Reroute

Fast reroute provides a mechanism for automatically rerouting traffic on an LSP if a node or link in an LSP fails, thus reducing the loss of packets traveling over the LSP. Fast rerouting is accomplished by precomputing and pre-establishing a number of detours along the LSP. Figure 13 illustrates an LSP from Router A to Router F, showing some of the detours that are established for the LSP. Each detour is established by an upstream node with the intent of avoiding the link toward the immediate downstream node and the immediate downstream node itself. Each detour might traverse through one or more label-switched routers that are not shown in the figure.

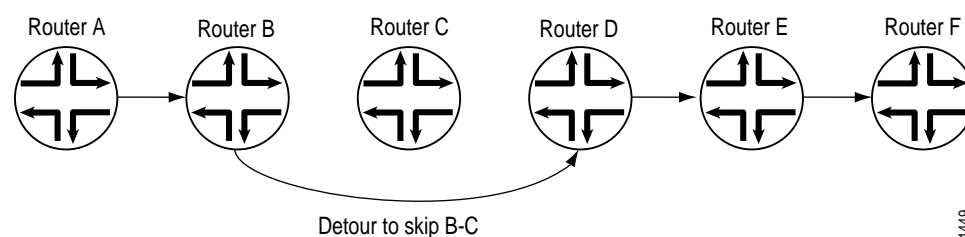
Figure 13: Detours Established for an LSP Using Fast Reroute



If a node detects either that a downstream link has failed (using a link-layer specific liveness detection mechanism) or that a downstream node has failed (for example, using the RSVP neighbor hello protocol), the node quickly splices the traffic onto the detour and, at the same time, signals the ingress router about the link or node failure. Figure 14 illustrates the detour taken when the link between Router B and Router C fails.

If the network topology is not rich enough, some of the detours might not succeed. For example, the detour from Router A to Router C in Figure 13 cannot traverse link A-B and Router B. If such a path is not possible, the detour does not occur.

Figure 14: Detour after the Link from Router B to Router C Fails



The time required for a fast-rerouting detour to take effect depends on two independent time intervals:

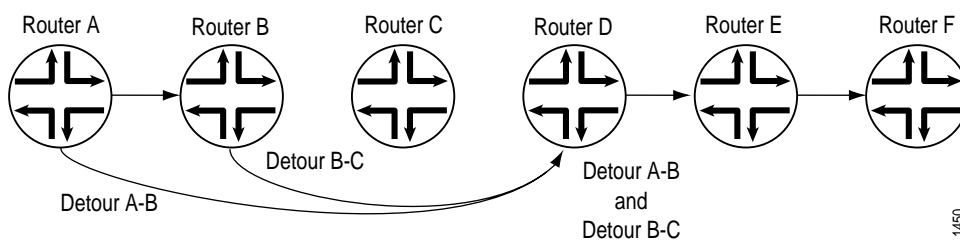
- Amount of time to detect that there is a link or node failure—This time interval depends greatly on the link layer in use and the nature of the failure. For example, failure detection on an SDH/SONET link typically is much faster than on a Gigabit Ethernet link, and both are much faster than detection of a router failure.
- Amount of time required to splice the traffic onto the detour—This time interval is primarily the CPU time required to update the routing table and then to update the forwarding table. The amount of time depends on the current CPU load and how busy the other routing protocols are that are sharing the CPU.

Fast reroute is a short-term patch to reduce packet loss. Because detour computation might not reserve adequate bandwidth, the detours might introduce congestion on the alternate links. The ingress router is the only router that is fully aware of LSP policy constraints and, therefore, is the only router able to come up with adequate long-term alternate paths.

Detours are created using RSVP and, like all RSVP sessions, they require extra state and overhead in the network. For this reason, each node establishes at most one detour for each LSP that has fast reroute enabled. Creating more than one detour for each LSP increases the overhead, but serves no practical purpose.

To reduce network overhead further, each detour attempts to merge back into the LSP as soon as possible after the failed node or link. If you can consider an LSP that travels through N router nodes, it is possible to create $N - 1$ detours. For instance, in Figure 15, the detour tries to merge back into the LSP at Router D instead of at Router E or Router F. Merging back into the LSP makes the detour scalability problem more manageable. If topology limitations prevent the detour from quickly merging back into the LSP, detours merge with other detours automatically.

Figure 15: Detours Merging into Other Detours



Fast reroute protects traffic against any single point of failure between the ingress and egress routers. If there are multiple failures along an LSP, it is possible that fast reroute itself might fail. Also, fast reroute does not protect against failure of the ingress or egress routers.

Computing and setting up detours are done independently at each node. On a node, if an LSP has fast reroute enabled and if a downstream link or node can be identified, the router performs a CSPF computation using the information in the local traffic engineering database (TED). For this reason, detours rely on your IGP's supporting traffic engineering extensions. Without the TED, detours cannot be established.

Detour computations might not succeed the first time. If a computation fails, the router recomputes detours approximately once every refresh interval until the computation succeeds.

To enable fast reroute on an LSP, include the `fast-reroute` statement at the `[edit protocols mpls label-switched-path lsp-path-name]` hierarchy level on the ingress router:

```
[edit protocols mpls label-switched-path lsp-path-name]
fast-reroute {
  bandwidth bps;
  (exclude group-names | no-exclude);
  hop-limit number;
  (include group-names | no-include);
}
```

You do not need to configure fast reroute on the LSP's transit and egress routers. Once fast reroute is enabled, the ingress router signals all the downstream routers that fast reroute is enabled on the LSP, and each downstream router does its best to set up detours for the LSP. If a downstream router does not support fast reroute, it ignores the request to set up detours and continues to support the LSP. A router that does not support fast reroute will cause some of the detours to fail, but otherwise has no impact on the LSP.

By default, no bandwidth is reserved for the rerouted path. To allocate bandwidth for the rerouted path, include the bandwidth statement. The bandwidth does not need to be identical to that allocated for the LSP.

Hop-limit constraints define how many more routers a detour is allowed to traverse compared to the LSP itself. By default, the hop limit is set to 6. For example, if an LSP traverses four routers, any detour for the LSP can be up to 10 (that is, 4 + 6) router hops, including the ingress and egress routers.

By default, a detour inherits the same administrative (coloring) group constraints as its parent LSP when CSPF is determining the alternate path. Administrative groups, also known as link coloring or resource class, are manually assigned attributes that describe the “color” of links, such that links with the same color conceptually belong to the same class. If you specify the include statement when configuring the parent LSP, all links traversed by the alternate session must have at least one color found in the list of groups. If you specify the exclude statement when configuring the parent LSP, all links must not have a color found in the list of groups. For more information about administrative group constraints, see “Configure Administrative Groups” on page 57.

Configure Addresses to Associate with the LSP

By default, a host route toward the egress router is installed in the inet.3 routing table. (The host route address is the one you configure in the to statement.) Installing the host route allows BGP to perform next-hop resolution. It also prevents the host route from interfering with prefixes learned from dynamic routing protocols and stored in the inet.0 routing table.

Unlike the routes in the inet.0 table, routes in the inet.3 table are not copied to the Packet Forwarding Engine, and hence they cause no changes in the system forwarding table directly. You cannot ping or traceroute through these routes. The only use for inet.3 is to permit BGP to perform next-hop resolution. To examine the inet.3 table, use the show route table inet.3 command.

To inject additional routes into the inet.3 routing table, include the install statement at the [edit protocols mpls label-switched-path *lsp-path-name*] hierarchy level:

```
[edit protocols mpls label-switched-path lsp-path-name]
install {
    destination/mask <active>;
}
```

The specified routes are installed as aliases into the routing table when the LSP is established. Installing additional routes allows BGP to resolve next hops within the specified prefix and to direct additional traffic for these next hops to a particular LSP.

Including the active option with the install statement installs the specified prefix into the inet.0 routing table, which is the primary forwarding table. The result is a route that is installed in the forwarding table any time the LSP is established, which means you can ping or traceroute the route. Use this option with care, because this type of prefix is very similar to a static route.

You use alias routes for routers that have multiple addresses being used as BGP next hops, or for routers that are not MPLS-capable. In either of these cases, the LSP can be configured to another MPLS-capable system within the local domain, which then acts as a “border” router. The LSP then terminates on the border router and, from that router, Layer 3 forwarding takes the packet to the true next-hop router.

In the case of an interconnect, the domain's border router can act as the proxy router and can advertise the prefix for the interconnect if the border router is not setting the BGP next hop to itself.

In the case of a POP that has routers that do not support MPLS, one router (for example, a core router) that supports MPLS can act as a proxy for the entire POP and can inject a set of prefixes that cover the POP. Thus, all routers within the POP can advertise themselves as IBGP next hops, and traffic can follow the LSP to reach the core router. This means that normal IGP routing would prevail within the POP.

You cannot use the ping or traceroute commands on routes in the inet.3 routing table.

For BGP next-hop resolution, it makes no difference whether a route is in inet.0 or inet.3; the route with the best match (longest mask) is chosen. Among multiple best-match routes, the one with the highest preference value is chosen.

Configure Path Connection Retry Information

The ingress router might make many attempts to connect and reconnect to the egress router using the primary path. You can control how often the ingress router tries to establish a connection using the primary path and how long it waits between retry attempts.

The retry timer configures how long the ingress router waits before trying to connect again to the egress router using the primary path. The default retry time is 30 seconds. The time can be from 1 through 600 seconds. To modify this value, include the retry-timer statement at the [edit protocols mpls label-switched-path *lsp-path-name*] hierarchy level:

```
[edit protocols mpls label-switched-path lsp-path-name]
  retry-timer seconds;
```

By default, no limit is set to the number of times an ingress router attempts to establish or re-establish a connection to the egress router using the primary path. To limit the number of attempts, include the retry-limit statement at the [edit protocols mpls label-switched-path *lsp-path-name*] hierarchy level:

```
[edit protocols mpls label-switched-path lsp-path-name]
  retry-limit number;
```

The limit can be a value up to 10000. When the retry limit is exceeded, no more attempts are made to establish a path connection. At this point, intervention is required to restart the primary path.

If you set a retry limit, it is reset to 1 each time a successful primary path is created.

Configure the Dynamic LSP Metric

If no specific metric is configured, an LSP attempts to track the IGP metric toward the same destination (the to address of the LSP). IGP includes OSPF, IS-IS, RIP, and static routes. BGP and other RSVP/LDP routes are excluded.

For example, if the OSPF metric toward a router is 20, all LSPs toward that router automatically inherit metric 20. If the OSPF toward a router later changes to a different value, all LSP metrics change accordingly. If there are no IGP routes toward the router, the LSP raises its metric to 65,535.

Note that in this case, the LSP metric is completely determined by IGP; it bears no relationship to the actual path the LSP is currently traversing. If LSP reroutes (such as through reoptimization), its metric does not change, and thus it remains transparent to users. Dynamic metric is the default behavior; no configuration is required.

Configure the Static LSP Metric

You can manually assign a fixed metric value to an LSP. Once configured using the metric statement at the [edit protocols mpls label-switched-path lsp-name] hierarchy level, the LSP metric is fixed and will not change:

```
[edit protocols mpls label-switched-path lsp-name]
metric number;
```

The LSP metric has several uses:

- When there are parallel LSPs with the same egress router, the metrics are compared to see which LSP has the lowest metric value (the lowest cost) and therefore the preferred path to the destination. If the metrics are the same, the traffic is shared.

Adjusting the metric values can force traffic to prefer some LSPs over others, regardless of the underlying IGP metric.

- When an IGP shortcut is enabled (see “IGP Shortcuts” on page 28), an IGP route might be installed in the routing table with an LSP as the next hop, if the LSP is on the shortest path to the destination. In this case, the LSP metric is added to the other IGP metrics to determine the total path metric. For example, if an LSP whose ingress router is X and egress router is Y is on the shortest path to destination Z, the LSP metric is added to the metric for the IGP route from Y to Z to determine the total cost of the path. If several LSPs are potential next hops, the total metrics of the paths are compared to determine which path is preferred (that is, has the lowest total metric). Or, IGP paths and LSPs leading to the same destination could be compared using the metric value to determine which path is preferred.

By adjusting the LSP metric, you can force traffic to prefer LSPs, to prefer the IGP path, or to share the load among them.

- If router X and Y are BGP peers, and if there is an LSP between them, the LSP metric represents the total cost to reach Y from X. If for any reason the LSP reroutes, the underlying path cost might change significantly, but X’s cost to reach Y remains the same (the LSP metric), which allows X to report through BGP MED a stable metric to downstream neighbors. As long as Y remains reachable through the LSP, no changes are visible to downstream BGP neighbors.

Configure CSPF Tie Breaking

When selecting a path for an LSP, CSPF uses a tie-breaking process if there are several equal-cost paths. For information about how CSPF selects a path, see “How CSPF Selects a Path” on page 26. To configure a random tie-breaking rule for CSPF to use to choose among equal-cost paths, include the random statement at the [edit protocols mpls path label-switched-path lsp-path-name]:

```
[edit protocols mpls path label-switched-path lsp-path-name]
random;
```


To prefer the path with the least utilized links, include the least-fill statement at the [edit protocols mpls path label-switched-path *lsp-path-name*]:

```
[edit protocols mpls path label-switched-path lsp-path-name]
least-fill;
```

To prefer the path with the most utilized links, include the most-fill statement at the [edit protocols mpls path label-switched-path *lsp-path-name*]:

```
[edit protocols mpls path label-switched-path lsp-path-name]
most-fill;
```

Configure Load-Balancing LSPs without CSPF

LSP tends to load-balance its placement by randomly selecting one of the equal-cost next hops and using it exclusively. The random selection is made independently at each transit router and is made by comparing IGP metrics alone. No consideration is given to bandwidth or congestion levels.

Disable Normal TTL Decrementing

By default, the TTL field value in the packet header is decremented by 1 for every hop the packet traverses in the LSP, thereby preventing loops. If the TTL field value reaches 0, packets are dropped, and an ICMP error packet might be sent to the originating router.

If normal TTL decrement is disabled, the TTL field of IP packets entering LSPs are decremented by only 1 upon transiting the LSP, making the LSP appear as a one-hop router to diagnostic tools, such as traceroute. This is done by the ingress router, which pushes a label on IP packets with the TTL field in the label initialized to 255. The label's TTL field value is decremented by 1 for every hop the MPLS packet traverses in the LSP. On the penultimate hop of the LSP, the router pops the label but does not write the label's TTL field value to the IP packet's TTL field. Instead, when the IP packet reaches the egress router, the IP packet's TTL field value is decremented by 1.

When you use traceroute to diagnose problems with an LSP, traceroute sees the ingress router, although the egress router performs the TTL decrement. Note that this assumes that traceroute is initiated outside of the LSP. The behavior of traceroute is different if it is initiated from the ingress router of the LSP. In this case, the egress router would be the first router to respond to traceroute.

You can disable normal TTL decrementing in an LSP so that the TTL field value does not reach 0 before the packet reaches its destination, thus preventing the packet from being dropped. You can also disable normal TTL decrementing to make the MPLS cloud appear as a single hop, thereby hiding the network topology.

There are two ways to disable TTL decrementing:

- On the ingress of the LSP, if you include the `no-decrement-ttl` statement at the `[edit protocols mpls label-switched-path lsp-path-name]` hierarchy level, the ingress router negotiates with all downstream routers using a proprietary RSVP object, to ensure all routers are in agreement. If negotiation succeeds, the whole LSP behaves as 1 hop to transit IP traffic.

```
[edit protocols mpls label-switched-path lsp-path-name]  
no-decrement-ttl;
```

Note that the RSVP object is proprietary to JUNOS software and might not work with other vendor software. Further, this potential incompatibility only applies to RSVP signaled LSPs, not LDP signaled LSPs. When you include the `no-decrement-ttl` statement, TTL hiding can be enforced on a per-LSP basis.

- On the router, you can include the `no-propagate-ttl` statement at the `[edit protocols mpls]` hierarchy level. This statement applies to all LSPs, regardless of whether they are RSVP-signaled or LDP-signaled. Once set, all future LSPs traversing through this router behave as a single hop to IP packets. LSPs established before you configure this statement are not affected.

```
[edit protocols mpls]  
no-propagate-ttl;
```

If you include the `no-propagate-ttl` statement, make sure all routers are configured consistently within an MPLS domain; failing to do so might cause the IP packet TTL to increase while in transit within LSPs. This can happen, for example, when the ingress router has `no-propagate-ttl` configured but the penultimate router does not, so the penultimate router writes the MPLS TTL value (which starts from the ingress router as 255) into the IP packet.

The operation of the `no-propagate-ttl` statement is more interoperable with other vendors' equipment. However, you must ensure all routers are configured identically.

Disable Constrained Path LSP Computation

If the IGP is a link state protocol and if it supports extensions that allow the current bandwidth reservation on each router's link to be reported, constrained path LSPs are computed by default.

The JUNOS implementations of IS-IS and OSPF include the extensions that support constrained-path LSP computation. In IS-IS, these extensions are enabled by default. (To disable this support, include the `disable` statement at the `[edit protocols isis traffic-engineering]` hierarchy level, as discussed in the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*. In OSPF, these extensions are disabled by default. To enable this support, include the `traffic-engineering` statement in the configurations of all routers running OSPF, as described in the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.)

If IS-IS is enabled on a router or if you enable OSPF traffic engineering extensions, MPLS performs the constrained-path LSP computation by default.

Constrained-path LSP computation works as follows: LSPs advertise their link information in the IGP's link-state packets. These packets are flooded throughout the network and hence provide information to all nodes. This link information is placed into the traffic engineering database (TED) and provides each ingress router with LSP topology information and recent LSP bandwidth reservation information. When computing complete paths for LSPs, the ingress router uses the information in the TED, along with the requirements you configure for the LSP, including bandwidth (configured with the `bandwidth` statement), hop limit (configured with the `hop-limit` statement), and the address of the egress router (configured with the `to` statement).

Constrained-path LSPs have a greater chance of being established quickly and successfully for several reasons:

- The LSP computation takes into account the current bandwidth reservation.
- Constrained-path LSPs reroute themselves away from node failures and congestion.

When constrained-path LSP computation is enabled, you can configure the LSP so that it is periodically re-optimized, as described in “Optimize Signaled LSPs” on page 63.

When an LSP is being established or when an existing LSP fails, the constrained-path LSP computation is repeated periodically at the interval specified by the retry timer, until the LSP is set up successfully. Once the LSP is set up, no recomputation is done. For more information about the retry timer, see “Configure Path Connection Retry Information” on page 53.

By default, constrained-path LSP computation is enabled. You might want to disable constrained-path LSP computation when all nodes do not support the necessary traffic engineering extensions. To disable constrained-path LSP computation, include the `no-cspf` statement at the `[edit protocols mpls label-switched-path lsp-path-name]` or `[edit protocols mpls label-switched-path lsp-path-name (primary | secondary)]` hierarchy level:

```
no-cspf;
```

Configure Administrative Groups

Administrative groups, also known as link coloring or resource class, are manually assigned attributes that describe the “color” of links, such that links with the same color conceptually belong to the same class. You can use administrative groups to implement a variety of policy-based LSP setups.

Administrative groups are meaningful only when constrained-path LSP computation is enabled.

Administrative groups require three levels of configuration. First, configure a table of group names at the `[edit protocols mpls]` hierarchy level:

```
[edit protocols mpls]
admin-groups{
  group-name group-value;
}
```

You can assign up to 32 names and values (in the range 0 through 31), which define a series of names and their corresponding values. The administrative names and values must be identical across all routers within a single domain.

To configure administrative groups, follow these steps:

1. Define multiple levels of service quality:

```
[edit]
protocols {
  mpls {
    admin-groups {
      best-effort 1;
      copper 2;
      silver 3;
      gold 4;
      violet 5;
    }
  }
}
```

2. Define administrative groups for an interface. These groups identify the administrative groups to which an interface belongs. You can assign multiple groups to an interface.

```
[edit]
protocols {
  mpls {
    interface interface name {
      admin-group [ group-name group-name... ];
    }
  }
}
```

If you do not include the `admin-group` statement, an interface does not belong to any group.

IGPs use the group information to build link-state packets, which are then flooded throughout the network, providing information to all nodes in the network. At any router, the IGP topology, as well as administrative groups of all the links, are available.

Changing the interface's administrative group affects only new LSPs. Existing LSPs on the interface are not preempted or recomputed to keep the network stable. If LSPs need to be removed because of a group change, issue the `clear RSVP session` command.

3. Configure an administrative group constraint for each LSP or for each primary or secondary LSP path, at the `[edit protocols mpls label-switched-path lsp-path-name]` or `[edit protocols mpls label-switched-path lsp-path-name (primary | secondary)]` hierarchy level:

```
[edit]
protocols {
  mpls {
    label-switched-path lsp-path-name {
      to address;
      ...
      primary path-name {
        admin-group {
          exclude [ group-name group-name ... ];
          include [ group-name group-name ... ];
        }
      }
    }
  }
}
```

```

secondary path-name {
  admin-group {
    exclude [ group-name group-name ... ];
    include [ group-name group-name ... ];
  }
}
admin-group {
  exclude [ group-name group-name ... ];
  include [ group-name group-name ... ];
}
}
}

```

If you omit the include or exclude statements, the path computation proceeds unchanged using constrained-path LSP computation. If you configure an exclude list, all chosen links must not have a color listed in the exclude list. If you configure an include list, all chosen links must have at least one color found in the include list. Links that have no color are automatically disqualified by any include or exclude list.



Note

Changing the LSP's administrative group causes an immediate recomputation of the route; therefore, the LSP might be rerouted.

Configure the LSP Preference

As an option, you can configure multiple LSPs between the same pair of ingress and egress routers. This is useful for balancing the load among the LSPs because all LSPs, by default, have the same preference level. To prefer one LSP over another, set different preference levels for individual LSPs. The LSP with the lowest preference value is used. The default preference of all LSPs is 7, which is lower (more preferred) than all learned routes except for direct interface routes.

To change the default preference value, include the preference statement at the [edit protocols mpls label-switched-path *lsp-path-name*] or [edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary)] hierarchy level:

```
preference preference;
```

Configure Whether to Record Path Routes

The JUNOS implementation of RSVP supports the Record Route Object, which allows an LSP to actively record the routers through which it transits. You can use this information for troubleshooting and to prevent routing loops. By default, path route information is recorded. To disable recording, include the no-record statement within the label-switched-path statement at the [edit protocols mpls label-switched-path *lsp-path-name*] or [edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary)] hierarchy level.

```
no-record;
```

Configure the MPLS CoS Value

When IP traffic enters an LSP tunnel, the ingress router marks all packets with a class-of-service (CoS) value, which is used to place the traffic into a transmission priority queue. On the router, for SDH/SONET and T3 interfaces, each interface has four transmit queues. The CoS value is encoded as part of the MPLS header and remains in the packets until the MPLS header is removed when the packets exit from the egress router. The routers within the LSP utilize the CoS value set at the ingress router.

MPLS class of service works in conjunction with the router's general CoS functionality. If you do not configure any CoS features, the default general CoS settings are used. For MPLS class of service, you might want to prioritize how the transmit queues are serviced by configuring weighted round-robin and to configure congestion avoidance using Random Early Detection (RED). The general CoS features are described in the *JUNOS Internet Software Configuration Guide: Interfaces and Chassis*.

When traffic enters an LSP tunnel, the CoS bits in the MPLS header are set in one of two ways. In the first way, the number of the output queue into which the packet was buffered and the PLP bit are written into the MPLS header and are used as the packet's CoS value. This behavior is the default, and no configuration is required. The *JUNOS Internet Software Configuration Guide: Interfaces and Chassis* explains the IP CoS values, and summarizes how the CoS bits are treated.

In the second way, you set a fixed CoS value on all packets entering the LSP tunnel. This means that all packets entering the LSP receive the same class of service. To do this, include the class-of-service statement at the [edit protocols mpls label-switched-path *lsp-path-name*] or [edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary)] hierarchy level

```
class-of-service cos-value;
```

The CoS value can be a decimal number from 0 through 7. This number corresponds to a 3-bit binary number. The high-order 2 bits of the CoS value select which transmit queue to use on the outbound interface card.

The low-order bit of the CoS value is treated as the packet loss priority (PLP) bit and is used to select the RED drop profile to use on the output queue. If the low-order bit is 0, the non-PLP drop profile is used, and if the low-order bit is 1, the PLP drop profile is used. It is generally expected that RED will more aggressively drop packets that have the PLP bit set. For more information about RED and drop profiles, see the *JUNOS Internet Software Configuration Guide: Interfaces and Chassis*.



Note

Configuring the PLP drop profile to drop packets more aggressively (for example, setting the CoS value from 6 to 7) decreases the likelihood of traffic getting through.

Table 1 summarizes how MPLS CoS values correspond to the transmit queue and PLP bit. Note that in MPLS, the mapping between the CoS bit value and the output queue is hard-coded. You cannot configure the mapping for MPLS; you can configure it only for IPv4 traffic flows, as described in the *JUNOS Internet Software Configuration Guide: Interfaces and Chassis*.

Table 1: MPLS CoS Values

MPLS CoS Value	Bits	Transmit Queue	PLP Bit
0	000	0	Not set
1	001	0	Set
2	010	1	Not set
3	011	1	Set
4	100	2	Not set
5	101	2	Set
6	110	3	Not set
7	111	3	Set

Because the CoS value is part of the MPLS header, the value is associated with the packets only as they travel through the LSP tunnel. The value is not copied back to the IP header when the packets exit from the LSP tunnel.

Configure an LSP to Be Adaptive

An LSP occasionally might need to reroute itself. Reasons include the following:

- Continuous reoptimization process is configured with the `optimize-timer` statement.
- The current path has connectivity problems.
- The LSP is preempted by another LSP configured with the `priority` statement and is forced to reroute.
- The explicit-path information for an active LSP is modified, or the LSP's bandwidth is increased.

You can configure an LSP to be *adaptive* when it is attempting to reroute itself. When it is adaptive, the LSP holds onto existing resources until the new path is successfully established and traffic has been cut over to the new LSP. To retain its resources, an adaptive LSP does the following:

- Maintains existing paths and allocated bandwidths—This ensures that the existing path is not torn down prematurely and allows the current traffic to continue flowing while the new path is being set up.
- Avoids double-counting for links that share the new and old paths—Double-counting occurs when an intermediate router does not recognize that the new and old paths belong to the same LSP and counts them as two separate LSPs, requiring separate bandwidth allocations. If some links are close to saturation, double-counting might cause the setup of the new path to fail.

By default, adaptive behavior is disabled. You can include the `adaptive` statement in two different hierarchy levels. If you specify the `adaptive` statement at the LSP hierarchy level [edit protocols mpls label-switched-path *lsp-path-name*], adaptive behavior is enabled on all primary/secondary paths of the LSP. This means both the primary and secondary paths share the same bandwidth on common links.

```
[edit protocols mpls label-switched-path lsp-path-name]
adaptive;
```

If you specify the adaptive statement at the primary/secondary hierarchy level [edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary)], adaptive behavior is enabled only on the path on which it is specified. Bandwidth double-counting happens between different paths.

```
[edit protocols mpls label-switched-path lsp-path-name (primary | secondary)]
adaptive;
```

Configure Priority and Preemption

When there is insufficient bandwidth to establish a more important LSP, you might want to tear down a less important existing LSP to free up the bandwidth. You do this by preempting the existing LSP.

Whether an LSP can be preempted is determined by two properties associated with the LSP:

- **Setup priority**—Determines whether a new LSP that preempts an existing LSP can be established. For preemption to occur, the setup priority of the new LSP must be higher than that of the existing LSP. Also, the act of preempting the existing LSP must produce sufficient bandwidth to support the new LSP. That is, preemption occurs only if the new LSP can be set up successfully.
- **Hold priority**—Determines the degree to which an LSP holds onto its session reservation after the LSP has been set up successfully. When the hold priority is high, the existing LSP is less likely to give up its reservation and hence it is unlikely that the LSP can be preempted.

You cannot configure an LSP with a high setup priority and a low hold priority because permanent preemption loops might result if two LSPs are allowed to preempt each other. You must configure the hold priority to be higher than or equal to the setup priority.

The setup priority also defines the relative importance of LSPs on the same ingress router. When the software starts, when a new LSP is established, or during fault recovery, the setup priority determines the order in which LSPs are serviced. Higher priority LSPs tend to be established first and hence enjoy more optimal path selection.

To configure the LSP's preemption properties, include the priority statement at the [edit protocols mpls label-switched-path *lsp-path-name*] or [edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary)] hierarchy level:

```
priority setup-priority hold-priority;
```

Both *setup-priority* and *hold-priority* can be a value from 0 through 7. The value 0 corresponds to the highest priority, and the value 7 to the lowest. By default, an LSP has a setup priority of 7 (that is, it cannot preempt any other LSPs) and a hold priority of 0 (that is, other LSPs cannot preempt it). These defaults are such that preemption does not happen. When you are configuring these values, the setup priority should always be less than or equal to the hold priority.

Optimize Signaled LSPs

Once an LSP has been established, topology or resources changes might, over time, make the path suboptimal. A subsequent recomputation might be able to determine a more optimal path.

If reoptimization is enabled, an LSP can be rerouted through different paths by constrained-path recomputations. However, if reoptimization is disabled, the LSP has a fixed path and cannot take advantage of newly available network resources. The LSP is fixed until the next topology change breaks the LSP and forces a recomputation.

Reoptimization is not related to failover. A new path is always computed when topology failures occur that disrupt an established path.

Because of the potential system overhead involved, you need to control carefully the frequency of reoptimization. Network stability might suffer when reoptimization is enabled. By default, `optimize-timer` is set to 0 (that is, it is disabled).

Configuring LSP optimization is meaningful only when constrained-path LSP computation is enabled, which is the default behavior. For more information about constrained-path LSP computation, see “Disable Constrained Path LSP Computation” on page 56.

To enable path reoptimization, include the `optimize-timer` statement at the [edit protocols mpls label-switched-path *lsp-path-name*] or [edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary)] hierarchy level:

```
optimize-timer seconds;
```

After reoptimization is run, the result is accepted only if it meets the following criteria:

1. The new path is not higher in IGP metric. (The metric for the old path is updated during computation, so if a recent link metric changed somewhere along the old path, it is accounted for.)
2. If the new path has the same IGP metric, it is not more hops away.
3. The new path does not cause preemption. (This is to reduce the ripple effect of preemption causing more preemption.)
4. The new path does not worsen congestion overall. This is done by comparing the percentage of available bandwidth on each link traversed by the new and old paths, starting from the most congested links.

When all the above conditions are met, then:

5. If the new path has a lower IGP metric, it is accepted.
6. If the new path has an equal IGP metric and lower hop count, it is accepted.
7. If you choose least-fill as a load-balancing algorithm and if the new path reduces congestion by at least 10 percent aggregated over all links it traversed, it is accepted. For random or most-fill algorithms, this rule does not apply.
8. Otherwise, the new path is rejected.

To disable items 2, 3, 4 and 6 above, enter the clear `mpls optimize-aggressive` command or at the `[edit protocols mpls]` hierarchy level, include the `optimize-aggressive` statement:

```
optimize-aggressive;
```

Including the `optimize-aggressive` statement makes the reoptimization process more aggressive. Not only does it tend to reroute more often, it also limits the reoptimization algorithm to be based on the IGP metric only.

Configure the Maximum Path Length

By default, each LSP can traverse a maximum of 255 hops, including the ingress and egress routers. To modify this value, include the `hop-limit` statement at the `[edit protocols mpls label-switched-path lsp-path-name]` or `[edit protocols mpls label-switched-path lsp-path-name (primary | secondary)]` hierarchy level:

```
hop-limit number;
```

The number of hops can be from 2 through 255. (A path with two hops consists of the ingress and egress routers only.)

Configure the Path Bandwidth

Each LSP has a bandwidth value. This value is included in the sender's `Tspec` field in RSVP path setup messages. To specify a bandwidth value, include the `bandwidth` statement at the `[edit protocols mpls label-switched-path lsp-path-name]` or `[edit protocols mpls label-switched-path lsp-path-name (primary | secondary)]` hierarchy level.

```
bandwidth bps;
```

You specify the bandwidth value in bits per second, with a higher value implying a greater user traffic volume. The default bandwidth is 0 bits per second.

A nonzero bandwidth requires transit routers to reserve capacity along the outbound links for the path. This is done using RSVP's reservation scheme. Any failure in bandwidth reservation (such as failures at RSVP policy control or admission control) might cause the LSP setup to fail.

Configure the Standby State

By default, secondary paths are set up only as needed. To have the system maintain a secondary path in a hot-standby state indefinitely, include the `standby` statement at the `[edit protocols mpls label-switched-path lsp-path-name secondary]` hierarchy level:

```
[edit protocols mpls label-switched-path lsp-path-name secondary]
standby;
```

The hot-standby state is meaningful only on secondary paths.

Maintaining a path in a hot-standby state enables swift cutover to the secondary path when downstream routers on the current active path indicate connectivity problems.

The hot-standby state has two advantages:

- It eliminates the call-setup delay during network topology changes. Call setup can suffer from significant delays when network failures trigger large numbers of LSP reroutes at the same time.
- A cutover to the secondary path can be made before RSVP learns that an LSP is down. There can be significant delays between the time the first failure is detected by protocol machinery (which can be an interface down, a neighbor becoming unreachable, a route becoming unreachable, or a transient routing loop being detected) and the time an LSP actually fails (which requires a timeout of soft state information between adjacent RSVP routers). When topology failures occur, hot-standby secondary paths can usually achieve the smallest cutover delays with minimal disruptions to user traffic.

When the primary path is considered to be stable again, traffic is automatically switched from the standby secondary path back to the primary path. The switch is performed no faster than twice the retry-timer interval and only if the primary path exhibits stability throughout the entire switch interval.

The drawback of the hot-standby state is that more state information must be maintained by all the routers along the path, which requires overhead from each of the routers.

Configure LSP Hold Time

When an LSP changes from being up to being down, or from down to up, this transition takes effect immediately in the router software and hardware. However, when advertising LSPs into IS-IS, you may want to damp LSP transitions, thereby not advertising the transition until a certain period of time has transpired (known as the hold time). In this case, if the LSP goes from up to down, the LSP is not advertised as being down until it has remained down for the hold-time period. Transitions from down to up are advertised into IS-IS immediately. Note that LSP damping only affects IS-IS advertisements of the LSP; other routing software and hardware react immediately to LSP transitions.

To damp LSP transitions, you can include the advertisement-hold-time statement at the [edit protocols mpls] hierarchy level:

```
[edit protocols mpls]
  advertise-hold-time seconds;
```

The *seconds* can be a value from 0 to 65535 seconds. The default is 5 seconds.

Configure LDP Tunneling

To correctly identify an LDP session associated with an RSVP LSP, ensure that the RSVP LSP endpoint address is the same as the transport address of the LDP peer.

Configure Alternate Backup Paths Using Fate Sharing

You can create a database of information that CSPF uses to compute one or more backup paths to use in case the primary path becomes unstable. The database describes the relationships between elements of the network, such as routers and links. Because these network elements share the same fate, this relationship is called *fate sharing*.

You can configure backup paths that minimize the number of shared links and fiber paths with the primary paths as much as possible to ensure that, if a fiber is cut, the minimum amount of data is lost and that a path still exists to the destination.

For a backup path to work optimally, it must not share links or physical fiber paths with the primary path. This ensures that a single point of failure will not affect the primary and backup paths at the same time.

To configure fate sharing, include the fate-sharing statement at the [edit routing-options] hierarchy level:

```
[edit routing-options]
fate-sharing {
  group group-name {
    cost value;
    from address <to address>;
  }
}
```

Each fate-sharing group must have a name, which can be up to 32 characters long and can contain letters, digits, periods (.) and hyphens (-). You can define up to 512 groups.

Fate-sharing groups contain three types of objects:

- Point-to-point links—Identified by the IP addresses at each end of the link. Unnumbered point-to-point links are typically identified by borrowing IP addresses from other interfaces. Order is not important; from 1.2.3.4 to 1.2.3.5 and from 1.2.3.5 to 1.2.3.4 have the same meaning.
- Nonpoint-to-point links—Include links on a LAN interface (such as Gigabit Ethernet interfaces), or NBMA interfaces, (such as ATM or Frame Relay). You identify these links by their individual interface address. For example, if a LAN 192.168.200.0/24 has four routers attached to it, each router link is individually identified:

```
from 192.168.200.1;   # LAN interface of router 1
from 192.168.200.2;   # LAN interface of router 2
from 192.168.200.3;   # LAN interface of router 3
from 192.168.200.4;   # LAN interface of router 4
```

Sequence is insignificant; you can list the addresses in any order.

- A router node—Identified by its configured router ID.

All objects in a group share certain similarities. For example, you can define a group for all fibers sharing the same fiber conduit, all optical channels that share the same fiber, all links that connect to the same LAN switch, all equipment sharing the same power source, and so on. All objects are treated as /32 host addresses.

For a group to be meaningful, it should contain at least two objects. You can configure groups with zero or one object; these groups are ignored during processing.

An object can be in any number of groups, and a group can contain any number of objects. Each group has a configurable cost attributed to it, which represents the level of impact this group has on CSPF computations. The higher the cost, the less likely a backup path will share with the primary path any objects in the group. The cost is directly comparable to traffic engineering metrics. By default, the cost is 1. Changing the fate-sharing database does not affect existing established LSPs until the next reoptimization of CSPF. The fate-sharing database does influence fast-reroute computations.

Implications to CSPF

When CSPF computes the primary paths of an LSP (or secondary paths when the primary path is not active), it ignores the fate-sharing information. You always want to find the best possible path (least IGP cost) for the primary path.

When CSPF computes a secondary path while the primary path (of the same LSP) is active, the following occurs:

1. CSPF identifies all fate-sharing groups that are associated with the primary path. CSPF does this by identifying all links and nodes that the primary path traverses and compiling group lists that contain at least one of the links or nodes. CSPF ignores the ingress and egress nodes in the search.
2. CSPF checks each link in the TED against the compiled group list. If the link is a member of a group, the cost of the link is increased by the cost of the group. If a link is a member of multiple groups, all group costs are added together.
3. CSPF performs the check for every node in the TED, except the ingress and egress node. Again, a node can belong to multiple groups, so costs are additive.
4. The router performs regular CSPF computation with the adjusted topology.

Example: Configure Fate Sharing

Configure fate-sharing groups thunder and shadow. Because shadow has no objects, it is ignored during processing.

```
[edit routing-options]
  fate-sharing {
    group thunder {
      cost 20;                # optional, default value is 1
      from 1.2.3.4 to 1.2.3.5; # a point-to-point link
      from 192.168.200.1;      # LAN interface
      from 192.168.200.2;      # LAN interface
      from 192.168.200.3;      # LAN interface
      from 192.168.200.4;      # LAN interface
      from 10.168.1.220;        # Router ID of a router node
      from 10.168.1.221;        # Router ID of a router node
    }
    group shadow {
    }
  }
```

Configure All Other MPLS Routers for Signaled LSPs

To configure signaled LSPs on all MPLS routers that should participate in MPLS, you need to enable MPLS and RSVP on these routers, as described in “Minimum MPLS Configuration” on page 41 and “Enable RSVP” on page 68.

Enable RSVP

For all routers that you want to have participate in signaled LSPs, you must enable RSVP because it is used to set up LSPs. To do this, include the following statements in the configuration. In general, we recommend that you enable RSVP on all router interfaces, except for those on the AS border:

```
[edit]
interfaces {
  interface-name {
    unit logical-unit-number {
      family mpls;
    }
  }
}
protocols {
  mpls {
    interface all;
  }
  rsvp {
    interface all;
  }
}
```

For more information about RSVP, see “RSVP Configuration Guidelines” on page 121.

Examples: Configure Signaled LSPs

On the ingress router, create a constrained path LSP in which the JUNOS software makes all the forwarding decisions. When the LSP is successfully set up, a route toward 11.1.1.1/32 is installed in the inet.3 table so that all BGP routes with matching BGP next-hop addresses can be forwarded through the LSP.

```
[edit]
interfaces {
  so-0/0/0 {
    unit 0 {
      family mpls;
    }
  }
}
protocols {
  rsvp {
    interface so-0/0/0;
  }
  mpls {
    label-switched-path to-hastings {
      to 11.1.1.1;
    }
    interface so-0/0/0;
  }
}
```

On the ingress router, create an explicit-path LSP and specify the transit routers between the ingress and egress routers. In this configuration, no constrained-path computation is performed. For the primary path, all intermediate hops are strictly specified so that its route cannot change. The secondary path must travel through router 14.1.1.1 first, then take whatever route is available to reach the destination. The remaining route taken by the secondary path is typically the shortest path computed by the IGP.

```
[edit]
interfaces {
  so-0/0/0 {
    unit 0 {
      family mpls;
    }
  }
}
protocols {
  rsvp {
    interface so-0/0/0;
  }
  mpls {
    path to-hastings {
      14.1.1.1 strict;
      13.1.1.1 strict;
      12.1.1.1 strict;
      11.1.1.1 strict;
    }
    path alt-hastings {
      14.1.1.1 strict;
      11.1.1.1 loose; # Any IGP route is acceptable
    }
    label-switched-path hastings {
      to 11.1.1.1;
      hop-limit 32;
      bandwidth 10m; # Reserve 10 mbps
      no-cspf;        # do not perform constrained-path computation
      primary to-hastings;
      secondary alt-hastings;
    }
    interface so-0/0/0;
  }
}
```

On the ingress router, create a constrained-path LSP in which the JUNOS software makes most of the forwarding decisions, taking into account the hop constraints listed in the path statements. The LSP is adaptive so that no bandwidth double-counting occurs on links shared by primary and secondary paths. To acquire the necessary link bandwidth, this LSP is allowed to preempt lower priority sessions. Finally, this path always keeps the secondary path in hot-standby state for quick failover.

```
[edit protocols]
mpls {
  path to-hastings {
    14.1.1.1 loose;
  }
  path alt-hastings {
    12.1.1.1 loose;
    11.1.1.1 strict;
  }
}
```

```

label-switched-path hasting {
  to 11.1.1.1;
  bandwidth 10m;      # Reserve 10 mbps
  priority 0 0;        # Preemptive, but not preemptable
  adaptive;            # Set adaptivity
  primary to-hastings;
  secondary alt-hastings {
    standby;
    bandwidth 1m;      # Reserve only 1 Mbps for the secondary path
  }
}
interface all;
}

```

On the ingress router, create a constrained-path LSP in which the JUNOS software makes most of the forwarding decisions for the primary path, subject to constraints of the path to-hastings, and in which the secondary path is an explicit path. The primary path must transit green or yellow links and must stay away from red links. The primary path is periodically recomputed and reoptimized. Finally, this path always keeps the secondary path in hot-standby state for quick failover.

When the LSP is up—either because the primary or secondary path is up, or both are up—the prefix 16.0.0.0/8 is installed in the inet.3 table so that all BGP routes whose BGP next hop falls within that range can use the LSP. Also the prefix 17/8 is installed in the inet.0 table so that BGP can only resolve its next hop through it and the route also can be reached using traceroute or ping. These two routes are in addition to the 11.1.1.1/32 route.

```

[edit protocols]
mpls {
  admin-groups {
    green 1;
    yellow 2;
    red 3;
  }
  path to-hastings {
    14.1.1.1 loose;
  }
  path alt-hastings {
    14.1.1.1 strict;
    13.1.1.1 strict;
    12.1.1.1 strict;
    11.1.1.1 strict;
  }
  label-switched-path hasting {
    to 11.1.1.1;
    bandwidth 100m;
    install 16.0.0.0/8;      # in inet.3; cannot use to traceroute or ping
    install 17.0.0.0/8 active; # installed in inet.0; can use to traceroute or ping
    primary to-hastings {
      admin-group {          # further constraints for path computation
        include [ green yellow ];
        exclude red;
      }
    }
    optimize-timer 3600;    # reoptimize every hour
  }
}

```



```

secondary alt-hastings {
    standby;
    no-cspf;          # do not perform constrained-path computation
}
}
interface all;
}

```

Configure MPLS over GRE Tunnels

MPLS LSPs can use GRE tunnels to cross routing areas, Autonomous Systems, and ISPs. Bridging MPLS LSPs over an intervening IP domain is possible without disrupting the outlying MPLS domain.

LSPs can reach any destination that the GRE tunnels can reach. MPLS applications can be deployed without requiring all transit nodes to support MPLS, or requiring all transit nodes to support the same label distribution protocols (LDP or RSVP). If you use CSPF, you must configure OSPF or IS-IS through the GRE tunnel. Traffic engineering is not supported over GRE tunnels; for example, you cannot reserve bandwidth or set priority or preemption.

For more information about GRE tunnels, see the *JUNOS Internet Software Configuration Guide: Interfaces, Class of Service, and Firewalls*.

Example: Configure MPLS over GRE Tunnels

To configure MPLS over GRE tunnels:

1. Enable family MPLS under the GRE interface configuration.

```

[edit interfaces]
interface gr-1/2/0 {
    unit 0 {
        tunnel {
            source 192.168.1.1;
            destination 192.168.1.2;
        }
        family inet {
            address 5.1.1.1/30;
        }
        family iso;
        family mpls;
    }
}

```

2. Enable RSVP and MPLS over the GRE tunnel.

```

[edit protocols]
rsvp {
    interface gr-1/2/0.0;
}
mpls {
    .....
    interface gr-1/2/0.0;
}
}

```

3. Configure LSPs to travel through the GRE tunnel end-point address.

```
[edit protocols]
mpls {
  label-switched-path gre-tunnel {
    to 5.1.1.2;
    .....
  }
}
```

Standard LSP configuration options apply. If the routing table specifies that a particular route will traverse a GRE tunnel, the RSVP packets will as well.

Chapter 6

Configure Static LSPs

To configure static LSPs, configure the ingress router and each router along the path up to and including the egress router.

For the ingress router, configure which packets to tag (based on the packet's IP destination address), the next router in the LSP, and the tag to apply to the packet. Manually assigned labels can have values in the range 16 through 1023. Optionally, you can apply preference and CoS values to the packets.

For the intermediate routers in the path, configure the next router in the path and the tag to apply to the packet. Again, you can optionally apply preference and CoS values to the packets.

For the egress router, you generally just remove the label and continue forwarding the packet to the next hop. However, if the previous router removed the label, the egress router examines the packet's IP header and forwards the packet toward its IP destination.

To configure static MPLS, perform the following tasks:

- Configure the Ingress Router for Static MPLS on page 73
- Configure the Intermediate and Egress Routers for Static MPLS on page 75

Configure the Ingress Router for Static MPLS

The ingress router checks the IP address in the incoming packet's destination address field and, if it finds a match in the routing table, applies the label associated with that address to the packets. The label has forwarding information associated with it, including the address of the next-hop router, and the route preference and CoS values.

To configure static LSPs on the ingress router, include the `static-path` statement at the [edit protocols mpls] hierarchy level:

```
[edit protocols mpls]
static-path inet {
  prefix {
    nexthop (address | interface-name | address/interface-name) # required
    push out-label; # required
    class-of-service value; # optional
    preference preference; # optional
  }
}
```

Each static-path statement consists of the following parts:

- Criteria to use to analyze an incoming packet:
 - The *inet* option creates an LSP that handles IPv4 packets. All static MPLS routes created using the *inet* option are installed in the default IPv4 routing table (*inet.0*) and the creating protocol is identified as static. This is no different from creating static IPv4 routes at the [edit routing-options static] hierarchy level.
 - In the *prefix* option, you configure the IP destination address to check when analyzing incoming packets. If the address matches, the specified label, *out-label*, is assigned to the packet and the packet enters an LSP. Each prefix that you specify is installed as a static route in the routing table. You can specify one or more *prefix* statements at the [edit protocols mpls static-path] hierarchy level.
- The *nexthop* statement supplies the IP address of the next hop to the destination. You can specify this as the IP address of the next hop, the interface name (for point-to-point interfaces only), or as *address/interface-name* to specify an IP address on an operational interface. When the next hop is on a directly attached interface, the route is installed in the routing table. You cannot configure a LAN or point-to-multipoint (NMBA) interface as a next-hop interface.
- Label properties applied to the packet in the LSP:
 - Label to apply to the packet (push *out-label*)—The label is a 20-bit integer, so it can be a number in the range 0 through 1048575 ($2^{20} - 1$). Dynamic MPLS assigns the labels 1024 through 1048575, so if your network uses both static and dynamic MPLS, we recommend that you use labels 16 through 1023 only for static MPLS. (Labels 0 through 15 are reserved.)
 - Preference of this route (preference *preference*).
 - CoS value to apply to the packet (class-of-service *cos-value*).

To determine whether a static ingress route is installed, use the command `show route table inet.0 protocol static`. The following is a sample output. The *push* keyword identifies that a label is to be added in front of IP packet.

```
10.0.0.0/8          *[Static/5] 00:01:48
                    > to 11.1.1.1 via so-0/0/0, push 123
```

Example: Configure the Ingress Router

Configure the ingress router for a static LSP that consists of three routers (see Figure 16). For packets addressed to 10.0.0.0/8, assign label 123 and transmit them to the next-hop router at 11.1.1.1:

```
interfaces {
  so-0/0/0 {
    unit 0 {
      family mpls;
    }
  }
}
```

```

protocols {
  mpls {
    static-path inet {
      10.0.0.0/8 {
        nexthop 11.1.1.1;
        push 123;
      }
    }
    interface so-0/0/0;
  }
}

```

To determine whether the static ingress route is installed, use the following command:

```
user@host> show route table inet.0 protocol static
```

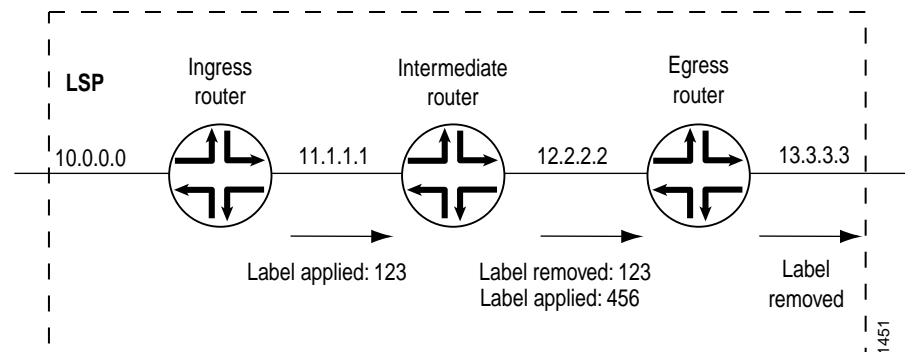
The following is a sample of the output. The push 123 keyword identifies the route.

```

10.0.0.0/8          *[Static/5] 00:01:48
                    > to 11.1.1.1 via so-0/0/0, push 123

```

Figure 16: Static MPLS Configuration



Configure the Intermediate and Egress Routers for Static MPLS

Intermediate and egress routers perform similar functions—they modify the label that has been applied to a packet. An intermediate router can change the label. An egress router removes the label (if the packet still contains a label) and continues forwarding the packet to its destination.

To configure static MPLS on intermediate and egress routers, include the interface statement at the [edit protocols mpls] hierarchy level:

```

[edit protocols mpls]
interface interface-name {
  label-map in-label {
    (nexthop <address; interface-name>) | (reject | discard); # Required
    (pop | (swap <out-label>)); # Required
    class-of-service value; # Optional
    preference preference; # Optional
    type type; # Optional
  }
}

```

Each statement within the interface statement consists of the following parts:

- Criteria to use to analyze the labeled packet. Two criteria are used: the interface on which the packet was received (specified in the opening interface statement itself) and the packet's label (specified in the label-map statement).
- The nexthop statement supplies the IP address of the next hop to the destination, specified as the IP address of the next hop, or the interface name (for point-to-point interfaces only), or *address/interface-name* to specify an IP address on an operational interface. When the specified next hop is on a directly attached interface, this route is installed in the routing table. You cannot configure a LAN or point-to-multipoint (NBMA) interface as a next-hop interface.
- Operation to perform on the labeled packet:
 - For egress routers, remove the packet's label altogether (pop).
 - For intermediate routers only, exchange the label for another label (swap *out-label*).
 - Discard the packet, sending an ICMP unreachable message to the packet's originator (reject).
 - Discard the packet without sending an ICMP unreachable message to the packet's originator (discard).
- Label properties to apply to the packet, all of which are optional:
 - Type of traffic in the LSP. Currently, the type can be IPv4 only (type *inet*), which is the default.
 - Preference value for this route (preference *preference*).
 - For intermediate routers only, the CoS value to apply to the packet (class-of-service *cos-value*).

You can specify any number of label-map statements at the [edit protocols mpls interface *interface-name*] hierarchy level.

The static routes are installed in the default MPLS routing table, mpls.0, and the creating protocol is identified as static. To verify that a static route is properly installed, use the command `show route table mpls.0 protocol static`. The following is an example of the output:

```
123                               *[Static/5] 00:00:38
                                > to 12.2.2.2 via so-5/0/0.0, swap 456
```

Example: Configure an Intermediate Router

For packets labeled 123 arriving on interface so-0/0/0, assign the label 456 and transmit them to the next-hop router at 12.2.2.2:

```
[edit]
interfaces {
  so-0/0/0 {
    unit 0 {
      family mpls;
    }
  }
}
protocols {
  mpls {
    interface so-0/0/0 {
      label-map 123 {
        nexthop 12.2.2.2;
        swap 456;
      }
    }
  }
}
```

To determine whether the static intermediate route is installed, use the following command:

```
user@host> show route table mpls.0 protocol static
```

The following is a sample of the output. The swap 456 keyword identifies the route.

```
123                               *[Static/5] 00:01:48
                                > to 12.2.2.2 via so-0/0/0, swap 456
```

Example: Configure an Egress Router

For packets labeled 456 arriving on interface so-0/0/0, remove the label and transmit the packets to the next-hop router at 13.3.3.3:

```
[edit]
interfaces {
  so-0/0/0 {
    unit 0 {
      family mpls;
    }
  }
}
protocols {
  mpls {
    interface so-0/0/0 {
      label-map 456 {
        nexthop 13.3.3.3;
        pop;
      }
    }
  }
}
```

To determine whether the static egress route is installed, use the following command:

```
user@host> show route table mpls.0 protocol static
```

The following is a sample of the output. The pop keyword identifies the egress route.

```
456                               *[Static/5] 00:01:48  
                                > to 13.3.3.3 via so-0/0/0, pop
```


Chapter 7

Configure Explicit-Path LSPs

If you disable constrained-path LSP computation, as described in “Disable Constrained Path LSP Computation” on page 56, you must configure LSPs manually. Experimenting with particular explicit paths can help you become familiar with MPLS.

When explicit-path LSPs are configured, the LSP is established along the path you specified. If the path is topologically not feasible, either because the network is partitioned or insufficient resources are available along some parts of the path, the LSP will fail. No alternative paths can be used. If the setup succeeds, the LSP stays on the defined path indefinitely.

To configure an explicit path LSP, follow these steps:

1. Configure the path information in a named path, as described in “Create a Named Path” on page 44. To configure complete path information, specify every router hop between the ingress and egress routers, preferably using the strict attribute. To configure incomplete path information, specify only a subset of router hops, using the loose attribute in places where the path is incomplete.

For incomplete paths, the MPLS routers complete the path by querying the local routing table. This query is done on a hop-by-hop basis, and each router can figure out only enough information to reach the next explicit hop. It might be necessary to traverse a number of routers in order to reach the next (loose) explicit hop.

Configuring incomplete path information creates portions of the path that are dependent on the current routing table, and this portion of the path can reroute itself as the topology changes. Therefore, an explicit-path LSP that contains incomplete path information is not completely fixed. These types of LSPs have only a limited ability to repair themselves, and they tend to create loops or flaps depending on the contents of the local routing table.

2. Configure the LSP and point it to the named path using either the primary or secondary statement, as described in “Configure the Primary and Secondary LSPs” on page 49.
3. Disable constrained-path LSP computation by including the no-cspf statement either as part of LSP or as part of a primary or secondary statement. For more information, see “Disable Constrained Path LSP Computation” on page 56.
4. Configure any other LSP properties.

Using explicit-path LSPs has the following drawbacks:

- More configuration effort is required.
- Configured path information cannot take into account dynamic network bandwidth reservation, so the LSPs tend to fail when resources become depleted.
- When an explicit-path LSP fails, you might need to manually repair it.

Because of these limitations, we recommend that you use explicit-path LSPs only in controlled situations, such as to enforce an optimized LSP placement strategy resulting from computations with an offline simulation software package.

Chapter 8

Configure Miscellaneous MPLS Properties

This chapter discusses the following topics:

- Configure Traffic Engineering for LSPs on page 81
- Configure MPLS to Gather Statistics on page 81
- Control MPLS System Log Messages and SNMP Traps on page 82
- Trace MPLS Protocol Packets and Operations on page 83

Configure Traffic Engineering for LSPs

Establishing an LSP installs a host route (a 32-bit mask) in the ingress router toward the egress router. The address of the host route is the destination address of the LSP. By default, only BGP can use LSPs in its route calculations. On the ingress router, to enable both BGP and the IGP to use an LSP in forwarding traffic destined for the egress router of that LSP, include the traffic-engineering statement at the [edit protocol mpls] hierarchy level:

```
[edit protocol mpls]
traffic-engineering bgp-igp;
```

Configure MPLS to Gather Statistics

You can configure MPLS so that it periodically gathers traffic statistics about all MPLS sessions, including transit sessions. To do this, include the statistics statement at the [edit protocol mpls] hierarchy level:

```
[edit protocol mpls]
statistics {
    file filename <size size files number>;
    interval seconds;
}
```

The default interval is 300 seconds.

The statistics are placed in a file, with one entry per LSP. At the end of each periodic report, a summary shows the current time, total number of sessions, numbers of sessions read, number of sessions ignored, and read errors, if any. Ignored sessions are typically those not in the up state or those with a reserved (0-15) incoming label (typically the egress point of an LSP). The reason for a read error appears on the same line as the entry for the LSP on which the error occurred. Gathering statistics is an unreliable process; occasional read errors might affect their accuracy. The following is a sample of the information included in the output file:

```
lsp6                0 pkt                0 Byte        0 pps        0 Bps        0%
lsp5                0 pkt                0 Byte        0 pps        0 Bps        0%
lsp6.1             34845 pkt            2926980 Byte   1049 pps     88179 Bps    132%
lsp5.1             0 pkt                0 Byte        0 pps        0 Bps        0%
lsp4               0 pkt                0 Byte        0 pps        0 Bps        0%
Dec  7 17:28:38 Total 6 sessions: 5 success, 0 fail, 1 ignored
```

Control MPLS System Log Messages and SNMP Traps

Whenever an LSP makes a transition from up to down, or down to up, and whenever an LSP switches from one active path to another, the ingress router generates a system log message and sends an SNMP trap. The following shows a sample system log message:

```
MPLS lsp sheep1 up on primary(any) Route  192.168.1.1 192.168.1.2 192.168.1.3
MPLS lsp sheep1 change on primary(any) Route  192.168.1.1 192.168.1.2
192.168.1.3
MPLS lsp sheep1 down on primary(any)
MPLS lsp sheep1 up on secondary(any) Route  192.168.1.1 192.168.1.2 192.168.1.3
MPLS lsp sheep1 change on secondary(any) to primary(any), Route  192.168.1.1
192.168.1.2 192.168.1.3
```

For information about the MPLS SNMP traps and the proprietary MPLS MIB, see the *JUNOS Internet Software Configuration Guide: Network Management*.

To disable both the generation of system log messages and SNMP traps, include the following log-updown statement at the [edit protocols mpls] hierarchy level:

```
[edit protocols mpls]
log-updown {
  no-syslog;
  no-trap;
}
```

To disable only the generation of system log messages, configure the following:

```
[edit]
user@host# set protocols mpls log-updown no-syslog
```

For scalability reasons, only the ingress router generates SNMP traps. By default, MPLS issues traps for all configured LSPs. If you have many LSPs, the number of traps can become quite large. To disable the generation of SNMP traps, configure the following:

```
[edit]
user@host# set protocols mpls log-updown no-trap
```

Trace MPLS Protocol Packets and Operations

To trace MPLS protocol packets and operations, include the `traceoptions` statement at the `[edit routing-options]` and `[edit protocol mpls]` hierarchy levels:

```
[edit protocol mpls]
traceoptions {
  file filename <replace> <size size> <files number> <no-stamp>
    <(world-readable | no-world-readable)>;
  flag flag <flag-modifier> <disable>;
}
```

You can specify the following MPLS-specific flags in the MPLS `traceoptions` statement:

- `connection`—Trace all circuit cross-connect (CCC) activity.
- `connection-detail`—Trace detailed CCC activity.
- `cspf`—Trace CSPF computations.
- `cspf-link`—Trace links visited during CSPF computations.
- `cspf-node`—Trace nodes visited during CSPF computations.
- `error`—Trace MPLS error conditions.
- `state`—Trace all LSP state transitions.

For general information about tracing and global tracing options, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Chapter 9

Summary of MPLS Configuration Statements

This chapter shows the complete MPLS configuration statements. The statements are organized alphabetically.

adaptive

Syntax	adaptive;
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>], [edit protocols mpls label-switched-path <i>lsp-path-name</i> (primary secondary) <i>path-name</i>]
Description	During reroute, do not double-count bandwidth on links shared by the old and new paths. Including this statement causes RSVP to use SE reservation styles and assists in smooth transition during rerouting. Do not use the adaptive and fast-reroute statements in the same LSP configuration because fast-reroute requires FF reservation styles.
Default	The configured object is disabled (operational).
Usage Guidelines	See “Configure an LSP to Be Adaptive” on page 61.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

admin-group

Syntax	admin-group [<i>group-names</i>];
Hierarchy Level	[edit protocols mpls interface <i>interface-name</i>]
Description	Define administrative groups for an interface.
Options	<i>group-names</i> —One or more names of groups defined with the admin-groups statement.
Usage Guidelines	See “Configure Administrative Groups” on page 57.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
See Also	admin-groups on page 86

Syntax	admin-group { include [group-names]; exclude [group-names]; }
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>], [edit protocols mpls label-switched-path <i>lsp-path-name</i> (primary secondary) path-name]
Description	Define the administrative groups to include or exclude for an LSP and for a path's primary and secondary paths.
Options	The statements are explained separately.
Usage Guidelines	See "Configure Administrative Groups" on page 57.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

admin-groups

Syntax	admin-groups { group-name group-value; }
Hierarchy Level	[edit protocols mpls]
Description	Configure administrative groups to implement link coloring or resource classes.
Options	<i>group-name</i> —Name of the group. You can assign up to 32 names. The names and their corresponding values must be identical across all routers within a single domain. <i>group-value</i> —Value assigned to the group. The names and their corresponding values must be identical across all routers within a single domain. Range: 0 through 31
Usage Guidelines	See "Configure Administrative Groups" on page 57.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
See Also	admin-group on page 85

advertise-hold-time

Syntax	advertise-hold-time <i>seconds</i> ;
Hierarchy Level	[edit protocols mpls]
Description	Do not advertise when the LSP goes from up to down, for a certain period of time known as hold time.
Options	<i>seconds</i> —Hold time specified in seconds. Range: 0 to 65535 seconds Default: 5 seconds

Usage Guidelines See “Configure LSP Hold Time” on page 65.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

bandwidth

Syntax bandwidth *bps*;

Hierarchy Level [edit protocols mpls label-switched-path *lsp-path-name*],
[edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary) *path-name*],
[edit protocols mpls label-switched-path *lsp-path-name* fast-reroute]

Description When configuring an LSP, specify the traffic rate associated with the LSP.

When configuring fast reroute, allocate bandwidth for the reroute path. By default, no bandwidth is reserved for the rerouted path. The fast reroute bandwidth does not need to be identical to that allocated for the LSP itself.

Options *bps*—Bandwidth specified in bits per second. You can specify this as an integer value (if you do so, count your zeros carefully, or you can use the abbreviations k (for a thousand), m (for a million), or g (for a billion [also called a thousand million]).
Range: Any positive integer
Default: 0 (no bandwidth is reserved)

Usage Guidelines See “Configure the Path Bandwidth” on page 64 and “Configure Fast Reroute” on page 49.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

class-of-service

Syntax class-of-service *cos-value*;

Hierarchy Level [edit protocols mpls label-switched-path *lsp-path-name*],
[edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary) *path-name*],
[edit protocols mpls interface *interface-name* label-map *in-label*],
[edit protocols mpls static-path inet *address*]

Description CoS value given to all packets in the LSP.

The CoS value might affect the scheduling or queuing algorithm of traffic traveling along an LSP.

Options *cos-value*—CoS value. A higher value typically corresponds to a higher level of service.
Range—0 through 7
Default—If you do not specify a CoS value, the IP precedence bits from the packet’s IP header are used as the packet’s CoS value.

Usage Guidelines See “Configure the MPLS CoS Value” on page 60, “Configure the Ingress Router for Static MPLS” on page 73, and “Configure the Intermediate and Egress Routers for Static MPLS” on page 75.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

disable

Syntax	disable;
Hierarchy Level	[edit protocols mpls], [edit protocols mpls interface <i>interface-name</i>], [edit protocols mpls label-switched-path <i>lsp-path-name</i>]
Description	Disable MPLS, an MPLS path, or an MPLS interface.
Default	The configured object is enabled (operational) unless explicitly disabled.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

discard

Syntax	discard;
Hierarchy Level	[edit protocols mpls interface <i>interface-name</i> label-map <i>in-label</i>]
Description	Do not forward packets that match the incoming label. Instead, drop the packets and do not send an ICMP unreachable message.
Usage Guidelines	See “Configure the Intermediate and Egress Routers for Static MPLS” on page 75.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

exclude

exclude (for administrative groups)

Syntax	exclude [<i>group-name</i>];
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i> admin-group], [edit protocols mpls label-switched-path <i>lsp-path-name</i> (primary secondary) <i>path-name</i> admin-group]
Description	Define the administrative groups to exclude for an LSP or for a path’s primary and secondary paths.
Options	<i>group-names</i> —One or more names of groups defined with the admin-groups statement.
Usage Guidelines	See “Configure Administrative Groups” on page 57.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

exclude (for fast reroute)

Syntax	(exclude [<i>group-name</i>] no-exclude);
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i> fast-reroute]
Description	Control exclusion of administrative groups: <ul style="list-style-type: none"> ■ exclude—Define the administrative groups to exclude for fast reroute. ■ no-exclude—Disable administrative group exclusion.
Options	<i>group-names</i> —One or more names of groups defined with the admin-groups statement.
Usage Guidelines	See “Configure Fast Reroute” on page 49.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

fast-reroute

Syntax	fast-reroute { bandwidth <i>bps</i> ; hop-limit <i>number</i> ; (include [<i>group-name</i>] no-include); (exclude [<i>group-name</i>] no-exclude); }
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>]
Description	Establish detours for the LSP so that if a node or link in the LSP fails, the traffic on the LSP can be rerouted with minimal packet loss. Do not use the adaptive and fast-reroute statements in the same LSP configuration because it can cause errors.
Options	The statements are explained separately.
Usage Guidelines	See “Configure Fast Reroute” on page 49.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

fate-sharing

Syntax	fate-sharing { group <i>group-name</i> { cost <i>value</i> ; from <i>address</i> <to <i>address</i> >; } }
Hierarchy Level	[edit routing-options]

Description Specify groups of objects that share certain similarities, resulting in backup paths to be used in case primary paths become unusable. All objects are treated as /32 host addresses. You specify one or more objects within a group. The objects can be LAN interfaces, router IDs, or point-to-point links. Sequence is insignificant.

Options group *group-name*—Each fate-sharing group must have a name, which can be up to 32 characters long and can contain letters, digits, periods (.) and hyphens (-). You can define up to 512 groups.

cost *value*—Cost assigned to the group.

Range: 1 through 65,535

Default: 1

from *address*—Address of ingress router.

to *address*—(Optional) Address of egress router. For point-to-point link objects, you must specify both a from and to address.

Usage Guidelines See “Configure Alternate Backup Paths Using Fate Sharing” on page 66.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

from

Syntax from *address*;

Hierarchy Level [edit protocols mpls label-switched-path *lsp-path-name*]

Description Specify the source address to use for the LSP.

The address you specify does not affect the outgoing interface used by the LSP.

Default If you do not include this statement, the software automatically selects the loopback interface as the address.

Options *address*—IP address.

Usage Guidelines See “Configure the Address of the Ingress Router” on page 48.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

hop-limit

Syntax	hop-limit <i>number</i> ;
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>], [edit protocols mpls label-switched-path <i>lsp-path-name</i> (primary secondary) <i>path-name</i>], [edit protocols mpls label-switched-path <i>lsp-path-name</i> fast-reroute]
Description	For an LSP, the maximum number of routers that the LSP can traverse, including the ingress and egress routers. For fast reroute, how many more routers a detour is allowed to traverse compared to the LSP itself. For example, if an LSP traverses four routers, any detour for the LSP can be no more than ten router hops, including the ingress and egress routers.
Options	<i>number</i> —Maximum number of hops. Range: 2 through 255 Default: 255 (for an LSP); 6 (for a detour)
Usage Guidelines	See “Configure Fast Reroute” on page 49 and “Configure the Maximum Path Length” on page 64.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

include

include (for administrative groups)

Syntax	include [<i>group-name</i>];
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i> admin-group], [edit protocols mpls label-switched-path <i>lsp-path-name</i> (primary secondary) <i>path-name</i> admin-group]
Description	Define the administrative groups to include for an LSP or for a path’s primary and secondary paths.
Options	<i>group-names</i> —One or more names of groups defined with the admin-groups statement.
Usage Guidelines	See “Configure Administrative Groups” on page 57.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

include (for fast reroute)

Syntax	(include [<i>group-names</i>] no-include);
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i> fast-reroute]
Description	Control inclusion of administrative groups: <ul style="list-style-type: none"> ■ include—Define the administrative groups to include for fast-reroute. ■ no-include—Disable administrative group inclusion.
Options	<i>group-names</i> —One or more names of groups defined with the admin-groups statement.
Usage Guidelines	See “Configure Fast Reroute” on page 49.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

install

Syntax	install { <i>destination-prefix/prefix-length</i> <active>; }
Hierarchy	[edit protocols mpls label-switched-path <i>lsp-path-name</i>]
Description	Associate one or more prefixes with an LSP. When the LSP is up, all the prefixes are installed as entries into the routing table.
Options	active —(Optional) Install the route into the forwarding table. Doing so allows you to issue a ping or traceroute command on this address. <i>destination-prefix/prefix-length</i> —Address to associate with the LSP.
Usage Guidelines	See “Configure Addresses to Associate with the LSP” on page 52.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

interface

Syntax	interface <i>interface-name</i> { disable; admin-group { <i>group-name</i> ; } label-map <i>in-label</i> { (nexthop (<i>address</i> <i>interface-name</i> <i>address/interface-name</i>)) (reject discard); (pop (swap < <i>out-label</i> >); class-of-service <i>value</i> ; preference <i>preference</i> ; type <i>type</i> ; } }
---------------	---

Hierarchy Level [edit protocols mpls]

Description Enable MPLS on one or more interfaces.

Options *interface-name*—Name of the interface on which to configure MPLS. To configure all interfaces, you can specify all. For details about specifying interfaces, see the *JUNOS Internet Software Configuration Guide: Interfaces, Class of Service, and Firewalls*.

The remaining options are explained separately in this chapter.

Usage Guidelines See “Minimum MPLS Configuration” on page 41 and “Configure the Intermediate and Egress Routers for Static MPLS” on page 75.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

label-map

Syntax label-map *in-label* {
 (nexthop (*address* | *interface-name* | *address/interface-name*)) | (reject | discard);
 (pop | (swap <*out-label*>);
 class-of-service *value*;
 preference *preference*;
 type *type*;
}

Hierarchy Level [edit protocols mpls interface *interface-name*]

Description For static MPLS only, the label to match.

Options *in-label*—Label value.
Range: 0 through 1048575. Dynamic MPLS assigns the labels 1024 through 1048575, so if your network uses both static and dynamic MPLS, we recommend that you use labels 16 through 1023 only for static MPLS. Labels 0 through 15 are reserved.

The remaining statements are explained separately in this chapter.

Usage Guidelines See “Minimum MPLS Configuration” on page 41 and “Configure the Intermediate and Egress Routers for Static MPLS” on page 75.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

label-switched-path

Syntax `label-switched-path lsp-path-name {`
`disable;`
`to address;`
`from address;`
`adaptive;`
`admin-group {`
`include group-names;`
`exclude group-names;`
`}`
`bandwidth bps;`
`class-of-service cos-value;`
`fast-reroute {`
`bandwidth bps;`
`hop-limit number;`
`(include group-names | no-include);`
`(exclude group-names | no-exclude);`
`}`
`hop-limit number;`
`ldp-tunneling;`
`metric number;`
`no-cspf;`
`no-decrement-ttl;`
`optimize-timer seconds;`
`preference preference;`
`priority setup-priority hold-priority;`
`(random | least-fill | most-fill);`
`(record | no-record);`
`retry-limit number;`
`retry-timer seconds;`
`standby;`
`primary path-name {`
`...`
`}`
`secondary path-name {`
`...`
`}`
`install {`
`destination/prefix-length <active>;`
`}`
`}`

Hierarchy Level [edit protocols mpls]

Description Configure an LSP to use in dynamic MPLS. When configuring an LSP, you must specify the address of the egress router in the to statement. All remaining statements are optional.

Options *lsp-path-name*—Name that identifies the LSP. The name can be up to 32 characters and can contain letters, digits, periods, and hyphens. To include other characters, enclose the name in quotation marks. The name must be unique within the ingress router.

The remaining statements are explained separately.

Usage Guidelines See “Create an LSP” on page 45.

Required Privilege Level routing—To view this statement in the configuration.
 routing-control—To add this statement to the configuration.

ldp-tunneling

Syntax	ldp-tunneling;
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>]
Description	Enable the LSP to be used for LDP tunneling.
Usage Guidelines	See “Configure LDP Tunneling” on page 65.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

least-fill

See random on page 103

log-updown

Syntax	log-updown { (syslog no-syslog); (trap no-trap); }
Hierarchy Level	[edit protocols mpls]
Description	Log a message or send a trap whenever an LSP makes a transition from up to down, or vice versa, and whenever an LSP switches from one active path to another. Only the ingress router performs these operations.
Default	When LSP transitions occur or paths switch, a message is logged to the system log file and an SNMP trap is sent.
Options	no-syslog —Do not log a message to the system log file. no-trap —Do not send an SNMP trap. syslog —Log a message to the system log file. trap —Send an SNMP trap. Default: syslog and trap
Usage Guidelines	See “Control MPLS System Log Messages and SNMP Traps” on page 82 and the <i>JUNOS Internet Software Configuration Guide: Network Management</i> .
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
See Also	traceoptions on page 108

metric

Syntax	metric <i>metric</i> ;
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>]
Description	Compare against another LSP or against an IGP route. To disable dynamic metric tracking, assign a fixed metric value to an LSP. If no metric is assigned, LSP metric is dynamic and automatically tracks underlying IGP metrics.
Options	<i>metric</i> —LSP metric value. Default: no metric assigned (dynamic) Range: 1 through 65535
Usage Guidelines	See “Configure the Dynamic LSP Metric” on page 53.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

most-fill

See random on page 103

mpls

Syntax	mpls { ... }
Hierarchy Level	[edit protocols]
Description	Enable MPLS on the router.
Usage Guidelines	See “Minimum MPLS Configuration” on page 41.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

nexthop

Syntax	nexthop (<i>address</i> <i>interface-name</i> <i>address/interface-name</i>);
Hierarchy Level	[edit protocols mpls interface <i>interface-name</i> label-map <i>in-label</i>], [edit protocols mpls static-path inet <i>address</i>]
Description	IP address of the next hop to the destination, specified as the IP address of the next hop, the interface name (for point-to-point interfaces only), and the <i>address/interface-name</i> to specify an IP address on an operational interface.
Options	<i>address</i> —IP address of the next-hop router. <i>interface-name</i> —IP address of the outgoing interface. It must be a point-to point interface. The name can be the simple name or a fully qualified domain name.

Usage Guidelines See “Configure the Ingress Router for Static MPLS” on page 73 and “Configure the Intermediate and Egress Routers for Static MPLS” on page 75.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-cspf

Syntax no-cspf;

Hierarchy Level [edit protocols mpls label-switched-path *lsp-path-name*],
[edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary) *path-name*]

Description Disable constrained-path LSP computation.

An explicit-path LSP is one that is completely configured through operator action. Once configured, it is initiated only along the explicitly specified path.

A constrained-path LSP relies on ingress router to compute the complete path. The ingress router takes into account the following information during the computation:

- IGP topology database
- Link utilization information from extensions in the IGP link-state database
- Administrative group information from extensions in the IGP link-state database
- LSP requirements, including bandwidth, hop count, and administrative group

Constrained-path LSPs can generally avoid link failures and congested links. They also permit recomputation (therefore, a new path) during topology changes or unsuccessful setup.

Default Constrained-path LSP computation enabled.

Usage Guidelines See “Configure Explicit-Path LSPs” on page 79.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-decrement-ttl

Syntax no-decrement-ttl;

Hierarchy Level [edit protocols mpls label-switched-path *lsp-path-name*]

Description Disable normal TTL decrementing, which decrements the TTL field value in the IP header by only 1 regardless of the number of label-switched routers in the LSP. The MPLS cloud appears as a single hop, thus hiding the network topology. The disable decrementing feature is valid only for RSVP-signaled LSPs.

Default Normal TTL decrementing enabled; the TTL field value is decremented by 1 as the packet passes through each label-switched router in the LSP.

Usage Guidelines See “Disable Normal TTL Decrementing” on page 55.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

See Also no-propagate-ttl on page 98

no-exclude

See exclude on page 88

no-include

See include on page 91

no-propagate-ttl

Syntax no-propagate-ttl;

Hierarchy Level [edit protocols mpls]

Description Disable normal TTL decrementing. You configure this statement once per router, and it affects all RSVP- or LDP-signaled LSPs. When this router acts as an ingress router for an LSP, it pushes an MPLS header with a TTL value of 255, regardless of the IP packet TTL. When the router acts as the penultimate router, it pops the MPLS header without writing the MPLS TTL into the IP packet.

Default Normal TTL decrementing enabled; the TTL field value is decremented by 1 as the packet passes through each label-switched router in the LSP.

Usage Guidelines See “Disable Normal TTL Decrementing” on page 55.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

See Also no-decrement-ttl on page 97

no-record

See record on page 103

optimize-aggressive

Syntax	optimize-aggressive;
Hierarchy Level	[edit protocols mpls]
Description	If enabled, the LSP reoptimization is based on the IGP metric alone, ignoring consideration of bandwidth, congestion, and hop-counts. This statement makes reoptimization more aggressive than the default.
Default	Aggressive optimization is disabled.
Usage Guidelines	See “Optimize Signaled LSPs” on page 63.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

optimize-timer

Syntax	optimize-timer <i>seconds</i> ;
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>], [edit protocols mpls label-switched-path <i>lsp-path-name</i> (primary secondary) <i>path-name</i>]
Description	<p>Enable periodic reoptimization of an LSP that is already set up. If topology changes occur, an existing path might become suboptimal, and a subsequent recomputation might be able to determine a better path. This option is useful only on LSPs for which constrained-path computation is enabled; that is, for which the no-cspf statement is not configured.</p> <p>To avoid extensive resource consumption that might result because of frequent path recomputations, or to avoid destabilizing the network as a result of constantly changing LSPs, we recommend that you either leave the timer value sufficiently large or disable the timer value.</p>
Default	The optimize timer is disabled.
Options	<i>seconds</i> —Length of the optimize timer. Range: 0 through 65535 Default: 0 seconds (the optimize timer is disabled)
Usage Guidelines	See “Optimize Signaled LSPs” on page 63.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

path

Syntax path *path-name* {
 address <strict | loose>
}

Hierarchy Level [edit protocols mpls]

Description Create a named path and optionally specify the sequence of explicit routers that form the path.

You must include this statement when configuring dynamic LSPs.

Options *address*—(Optional) IP address of each transit router in the LSP. You must specify the address or host name of each transit router, although you do not need to list each transit router if its type is loose. As an option, you can include the ingress and egress routers in the path. Specify the addresses in order, starting with the ingress router (optional) or the first transit router, and continuing sequentially along the path until reaching the egress router (optional) or the router immediately before the egress router.
Default: If you do not specify any routers explicitly, no routing limitations are imposed on the LSP.

loose—(Optional) Indicate that the next address in the path statement is a loose link. This means that the LSP can traverse through other routers before reaching this router.

Default: strict

path-name—Name that identifies the sequence of nodes that form an LSP. The name can be up to 16 characters and can contain letters, digits, periods, and hyphens. To include other characters or use a longer name, enclose the name in quotation marks. The name must be unique within the ingress router.

strict—(Optional) Indicate that the LSP must go to the next address specified in the path statement without traversing other nodes. This is the default.

Usage Guidelines See “Create a Named Path” on page 44.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

See Also static-path on page 106

pop

Syntax pop;

Hierarchy Level [edit protocols mpls interface *interface-name* label-map *in-label*]

Description Remove the label from the top of the label stack. If there is another label in the stack, that label becomes the label at the top of the label stack. Otherwise, the packet is forwarded as a native protocol packet (typically, as an IP packet).

Usage Guidelines See “Configure the Intermediate and Egress Routers for Static MPLS” on page 75.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

preference

Syntax	<code>preference <i>preference</i>;</code>
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>], [edit protocols mpls label-switched-path <i>lsp-path-name</i> (primary secondary) <i>path-name</i>], [edit protocols mpls interface <i>interface-name</i> label-map <i>in-label</i>], [edit protocols mpls static-path inet <i>address</i>]
Description	<p>Preference for the route.</p> <p>You can optionally configure multiple LSPs between the same pair of ingress and egress routers. This is useful for balancing the load among the LSPs because all LSPs, by default, have the same preference level. To prefer one LSP over another, set different preference levels for individual LSPs. The LSP with the lowest preference value is used. The default preference of all LSPs is 7, which is lower (more preferred) than all learned routes except for direct interface routes.</p>
Options	<p><i>preference</i>—Preference to assign to the route. A route with a lower preference value is preferred.</p> <p>Range: 1 through 255</p> <p>Default: 7</p>
Usage Guidelines	See “Configure the Ingress Router for Static MPLS” on page 73 and “Configure the Intermediate and Egress Routers for Static MPLS” on page 75.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

primary

Syntax	<pre>primary <i>path-name</i> { adaptive; admin-group { include <i>group-names</i>; exclude <i>group-names</i>; } bandwidth <i>bps</i>; class-of-service <i>cos-value</i>; hop-limit <i>number</i>; no-cspf; optimize-timer <i>seconds</i>; preference <i>preference</i>; priority <i>setup-priority hold-priority</i>; (record no-record); standby; }</pre>
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>]
Description	<p>Specify the primary path to use for an LSP. You can configure only one primary path.</p> <p>You can optionally specify preference, CoS, and bandwidth values for the primary path, which override any equivalent values that you configure for the LSP (at the [edit mpls label-switched-path <i>lsp-path-name</i>] hierarchy level).</p>

Options *path-name*—Name of a path that you created with the path statement.

The remaining statements are explained separately.

Usage Guidelines See “Configure the Primary and Secondary LSPs” on page 49.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

priority

Syntax *priority setup-priority hold-priority;*

Hierarchy Level [edit protocols mpls label-switched-path *lsp-path-name*],
[edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary) *path-name*]

Description If, at session setup time, insufficient link bandwidth is encountered during session establishment, the setup priority is compared with existing established sessions on the link to determine whether some of them should be preempted to accommodate the new session. For a session to be preempted, its hold priority must be lower.

Options *setup-priority*—Setup priority.
Range: 0 through 7, where 0 is the highest and 7 is the lowest priority
Default: 7 (The session cannot preempt any existing sessions.)

hold-priority—Hold priority, which is used to keep a reservation after it has been set up. A smaller number has a higher priority. The priority must be greater than or equal to the setup priority to prevent preemption loops.
Range: 0 through 7 (0 is the highest and 7 is the lowest priority.)
Default: 0 (Once the session is set up, no other session can preempt it.)

Usage Guidelines See “Configure Priority and Preemption” on page 62.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

push

Syntax *push out-label;*

Hierarchy Level [edit protocols mpls static-path inet *address*]

Description Add a new label to the top of the label stack.

Options *out-label*—Label value.
Range: 0 through 1048575 (Dynamic MPLS assigns the labels 1024 through 1048575, so if your network uses both static and dynamic MPLS, we recommend that you use labels 16 through 1023 only for static MPLS. Labels 0 through 15 are reserved.)

Usage Guidelines See “Configure the Intermediate and Egress Routers for Static MPLS” on page 75.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

random

Syntax	(random least-fill most-fill);
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>]
Description	<p>Configure the preferred path when several equal-cost candidate paths to a destination exist, and prefer the path with the highest available bandwidth (with the largest minimum available bandwidth ratio). The available bandwidth ratio of a link is the available bandwidth on a link divided by the maximum reservable bandwidth on the link.</p> <ul style="list-style-type: none"> ■ least-fill—Prefer the path with the most available bandwidth (with the largest minimum available bandwidth ratio). ■ most-fill—Prefer the path with the least available bandwidth (with the minimum available bandwidth ratio). The minimum available bandwidth ratio of a path is the smallest available bandwidth ratio belonging to any of the links in the path. ■ random—Choose the path at random. <p>Default: random</p>
Usage Guidelines	See “Configure CSPF Tie Breaking” on page 54.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

record

Syntax	(record no-record);
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>], [edit protocols mpls label-switched-path <i>lsp-path-name</i> (primary secondary) <i>path-name</i>]
Description	Specify whether an LSP should actively record the routes in the path. Recording routes requires that all transit routers support the RSVP Record Route Object. Recording routes can be useful for diagnostics and loop detection.
Default	Record routes.
Usage Guidelines	See “Configure Whether to Record Path Routes” on page 59.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

reject

Syntax	reject;
Hierarchy Level	[edit protocols mpls interface <i>interface-name</i> label-map <i>in-label</i>]
Description	Do not forward a packet with the matching incoming label. Instead, drop the packet and, for IP packets, send an ICMP unreachable message to the packet's originator.
Usage Guidelines	See "Configure the Intermediate and Egress Routers for Static MPLS" on page 75.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

retry-limit

Syntax	retry-limit <i>number</i> ;
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>], [edit protocols mpls label-switched-path <i>lsp-path-name</i> (primary secondary) <i>path-name</i>]
Description	Maximum number of times the ingress router tries to establish the primary path. This counter is reset each time a primary path is created successfully. When the limit is exceeded, no more connection attempts are made. Intervention is then required to restart the connection.
Options	<i>number</i> —Maximum number of tries to establish the primary path. Range: 0 through 10000 Default: 0 (The ingress node never stops trying to establish the primary path.)
Usage Guidelines	See "Configure Path Connection Retry Information" on page 53.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

retry-timer

Syntax	retry-timer <i>seconds</i> ;
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>], [edit protocols mpls label-switched-path <i>lsp-path-name</i> (primary secondary) <i>path-name</i>]
Description	Amount of time the ingress router waits between attempts to establish the primary path.
Options	<i>seconds</i> —Amount of time between attempts to connect to the primary path. Range: 1 through 600 Default: 30 seconds
Usage Guidelines	See "Configure Path Connection Retry Information" on page 53.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

secondary

Syntax `secondary path-name {
 adaptive;
 admin-group {
 include group-names;
 exclude group-names;
 }
 bandwidth bps;
 class-of-service cos-value;
 hop-limit number;
 no-cspf;
 optimize-timer seconds;
 preference preference;
 priority setup-priority reservation-priority;
 (record | no-record);
 standby;
}`

Hierarchy Level [edit protocols mpls label-switched-path *lsp-path-name*]

Description Specify one or more secondary paths to use for the LSP. You can configure more than one secondary path. All secondary paths are equal, and the first one that is available is chosen.

You can specify secondary paths even if you have not specified any primary paths.

Optionally, you can specify preference, CoS, and bandwidth values for the secondary path, which override any equivalent values that you configure for the LSP (at the [edit mpls label-switched-path] hierarchy level).

Options *path-name*—Name of a path that you created with the path statement.

The remaining statements are explained separately.

Usage Guidelines See “Configure the Primary and Secondary LSPs” on page 49.

Required Privilege Level routing—To view this statement in the configuration.
 routing-control—To add this statement to the configuration.

standby

Syntax `standby;`

Hierarchy Level [edit protocols mpls label-switched-path *lsp-path-name*],
 [edit protocols mpls label-switched-path *lsp-path-name* (primary | secondary) *path-name*]

Description Have the path remain up at all times to provide instant switchover if connectivity problems occur.

Usage Guidelines See “Configure the Standby State” on page 64.

Required Privilege Level routing—To view this statement in the configuration.
 routing-control—To add this statement to the configuration.

static-path

Syntax static-path inet {
 prefix {
 nexthop *address*;
 push *out-label*;
 preference *preference*;
 class-of-service *value*;
 }
 }

Hierarchy Level [edit protocols mpls]

Description Statically configure an LSP. You configure the LSP on the ingress router only.

You can specify one or more static-path statements.

Options *prefix*—IP address that matches the packet's destination field. You can specify this option in one of the following ways:

■ IP address—Example: 10.0.0.2

■ Range of IP addresses—Example: 10.0.0.0/8

You can specify one or more addresses.

inet—Configure the path for packets with IPv4 destinations.

The remaining options are explained separately.

Usage Guidelines See “Configure the Ingress Router for Static MPLS” on page 73.

Required Privilege Level routing—To view this statement in the configuration.
 routing-control—To add this statement to the configuration.

statistics

Syntax statistics {
 file *filename* <*size size* files *number*>;
 interval *seconds*;
 }

Hierarchy Level [edit protocols mpls]

Description Enable MPLS statistics collection and reporting.

Options file *filename*—Name of the file to receive the output. We recommend that you place MPLS tracing output in the file mpls-stat in the /var/log directory.

files *number*—(Optional) Maximum number of trace files. When a trace file named *file* reaches its maximum size, it is renamed *file.0*, then *file.1*, and so on, until the maximum number of files is reached. Then, the oldest file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the size option.

Range: 2 or more

Default: 2 files

interval *seconds*—(Optional) Interval at which to periodically collect statistics.

Range: 1 through 65535

Default: 300 seconds

size *size*—(Optional) Maximum size of each file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a file named *file* reaches this size, it is renamed *file.0*. When the *file* again reaches its maximum size, *file.0* is renamed *file.1* and *file* is renamed *file.0*. This renaming scheme continues until the maximum number of files is reached. Then, the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of files with the files option.

Syntax: *xk* to specify KB, *xm* to specify MB, or *xg* to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

Usage Guidelines See “Configure MPLS to Gather Statistics” on page 81.

Required Privilege Level routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

swap

Syntax swap <*out-label*>;

Hierarchy Level [edit protocols mpls interface *interface-name* label-map *in-label*]

Description Remove the label at the top of the label stack and replace it with the specified label.

Options *out-label*—(Optional) Label value.

Range: 0 through 1048575 (Dynamic MPLS assigns the labels 1024 through 1048575, so if your network uses both static and dynamic MPLS, we recommend that you use labels 16 through 1023 only for static MPLS. Labels 0 through 15 are reserved.)

Default: If you omit *out-label*, the original label value remains unchanged.

Usage Guidelines See “Configure the Intermediate and Egress Routers for Static MPLS” on page 75.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

to

Syntax	<code>to address;</code>
Hierarchy Level	[edit protocols mpls label-switched-path <i>lsp-path-name</i>]
Description	Specify the egress router of a dynamic LSP.
Options	<i>address</i> —Address of the egress router.
Usage Guidelines	See “Configure the Address of the Egress Router” on page 48.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

traceoptions

Syntax	<pre> traceoptions { file <i>filename</i> <replace> <size <i>size</i>> <files <i>number</i>> <no-stamp> <(world-readable no-world-readable)>; flag <i>flag</i> <<i>flag-modifier</i>> <disable>; } </pre>
Hierarchy Level	[edit protocols mpls]
Description	<p>Configure MPLS protocol-level tracing options.</p> <p>To specify more than one tracing operation, include multiple flag statements.</p>
Default	The default MPLS protocol-level tracing options are those inherited from the routing protocols traceoptions statement included at the [edit routing-options] hierarchy level.
Options	<p>disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.</p> <p><i>filename</i>—Name of the file to receive the output of the tracing operation. All files are placed in the directory <code>/var/log</code>. We recommend that you place MPLS tracing output in the file <code>mpls-log</code>.</p> <p><i>files number</i>—(Optional) Maximum number of trace files. When a trace file named <i>trace-file</i> reaches its maximum size, it is renamed as <i>trace-file.0</i>, then as <i>trace-file.1</i>, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.</p> <p>If you specify a maximum number of files, you also must specify a maximum file size with the size option.</p> <p>Range: 2 to 1000 Default: 2 files</p>

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements.

MPLS Tracing Flags

- connection—All circuit cross-connect (CCC) activity
- connection-detail—Detailed CCC activity
- cspf—CSPF computations
- cspf-link—Links visited during CSPF computations
- cspf-node—Nodes visited during CSPF computations
- error—MPLS error packets
- state—All LSP state transitions

Global Tracing Flags

- all—All tracing operations
- general—A combination of the normal and route trace operations
- normal—All normal operations
Default: If you do not specify this option, only unusual or abnormal operations are traced.
- policy—Policy operations and actions
- route—Routing table changes
- state—State transitions
- task—Interface transactions and processing
- timer—Timer usage

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- detail—Detailed trace information
- receive—Packets being received
- send—Packets being transmitted

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Disallow any user to read the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file.0*. When the *trace-file* again reaches its maximum size, *trace-file.0* is renamed *trace-file.1* and *trace-file* is renamed *trace-file.0*. This renaming scheme continues until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the files option.

Syntax: *xk* to specify KB, *xm* to specify MB, or *xg* to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Usage Guidelines See “Trace MPLS Protocol Packets and Operations” on page 83.

Required Privilege Level routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

traffic-engineering

Syntax traffic-engineering (bgp | bgp-igp);

Hierarchy Level [edit protocols mpls]

Description Select whether MPLS performs traffic engineering on BGP destinations only or on both BGP and IGP destinations. Affects only LSPs originating from this router, not transit or egress LSPs.

Options bgp—on BGP destinations only. Ingress routes are installed in the inet.3 routing table.

bgp-igp—on both BGP and IGP destinations. Ingress routes are installed in the inet.0 routing table. If IGP shortcuts are enabled, the shortcut routes are therefore automatically installed in the inet.0 routing table.

Default: bgp

Usage Guidelines See “Configure Traffic Engineering for LSPs” on page 81.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

type

Syntax	<code>type type;</code>
Hierarchy Level	[edit protocols mpls interface <i>interface-name</i> label-map <i>in-label</i>]
Description	Type of traffic in the LSP.
Options	<i>type</i> —Traffic type. It can be inet (for IPv4 traffic).
Usage Guidelines	See “Configure the Intermediate and Egress Routers for Static MPLS” on page 75.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

type

.....

Part 3

RSVP

- RSVP Overview on page 115
- RSVP Configuration Guidelines on page 121
- Summary of RSVP Configuration Statements on page 129

Chapter 10

RSVP Overview

This chapter discusses the following topics:

- RSVP Overview on page 115
- RSVP Standards on page 116
- JUNOS RSVP Protocol Implementation on page 116
- RSVP Operation on page 117
- RSVP Message Types on page 117
- RSVP Reservation Styles on page 119

RSVP Overview

RSVP is a resource reservation setup protocol that is used by both network hosts and routers. Hosts use RSVP to request a specific quality of service (QoS) from the network for particular application flows. Routers use RSVP to deliver QoS requests to all routers along the data path. RSVP also can maintain and refresh states for a requested QoS application flow.

RSVP treats an application flow as a simplex connection. That is, the QoS request travels only in one direction—from the sender to the receiver. RSVP is a transport layer protocol that uses IP as its network layer. However, RSVP does not transport application flows. Rather, it is more of an Internet control protocol, similar to ICMP, IGMP, IS-IS, or OSPF. RSVP runs as a separate software process in the JUNOS Internet software and is not in the packet forwarding path.

RSVP is not a routing protocol, but rather is designed to operate with current and future unicast and multicast routing protocols. The routing protocols are responsible for choosing the routes to use to forward packets, and RSVP consults local routing tables to obtain routes. RSVP is responsible only for ensuring the QoS of packets traveling along a data path.

The receiver in an application flow is responsible for requesting the preferred QoS from the sender. To do this, the receiver issues an RSVP QoS request on behalf of the local application. The request propagates to all routers in reverse direction of the data paths toward the sender. In this process, RSVP requests might be merged, resulting in a protocol that scales well when there are a large number of receivers.

Because the number of receivers in an application flow is likely to change, and the flow of delivery paths might change during the life of an application flow, RSVP takes a soft-state approach in its design, creating and removing the protocol states in routers and hosts incrementally over time. RSVP sends periodic refresh messages to maintain its state and to recover from occasional lost messages. In the absence of refresh messages, the RSVP states automatically time out and are deleted.

RSVP Standards

RSVP is described in several RFC draft documents. The following documents provide a good overview of RSVP:

- RFC 2205, *Resource Reservation Protocol (RSVP), Version 1, Functional Specification*
- RFC 2209, *Resource Reservation Protocol (RSVP), Version 1, Message Processing Rules*
- RFC 2210, *The Use of RSVP with IETF Integrated Services*
- RFC 2211, *Specification of the Controlled-Load Network Element Service*
- RFC 2215, *General Characterization Parameters for Integrated Service Network Elements*
- RFC 2216, *Network Element Service Specification Template*
- *Extensions to RSVP for LSP Tunnels*, Internet draft draft-ietf-mpls-rsvp-lsp-tunnel-05.txt
- *RFC 2747, RSVP Cryptographic Authentication*
- *RSVP Refresh Reduction Extensions*, Internet draft draft-ietf-rsvp-refresh-reduct-05.txt

To access Internet RFCs and drafts, go to the IETF Web site at <http://www.ietf.org>.

JUNOS RSVP Protocol Implementation

The JUNOS implementation of RSVP supports RSVP Version 1. The software includes support for all mandatory objects and RSVP message types, and supports message integrity and node authentications through the Integrity Object.

The primary purpose of the JUNOS RSVP software is to support dynamic signaling within MPLS label-switched paths. Supporting resource reservations over the Internet is only a secondary purpose of the JUNOS implementation. Because of this, the RSVP software does not support the following features:

- IP multicasting sessions.
- Traffic control—It cannot make resource reservations for real-time video or audio sessions.

With regard to the protocol mechanism, packet processing, and RSVP objects supported, the JUNOS implementation of the software is interoperable with other RSVP implementations.

RSVP Operation

RSVP creates independent sessions to handle each data flow. A session is identified by a combination of the destination address, an optional destination port, and a protocol. Within a session, there can be one or more senders. Each sender is identified by a combination of its source address and source port. An out-of-band mechanism, such as a session announcement protocol or human communication, is used to communicate the session identifier to all senders and receivers.

A typical RSVP session involves the following sequence of events:

1. A potential sender starts sending RSVP Path messages to the session address.
2. A receiver, wanting to join the session, registers itself if necessary. For example, a receiver in a multicast application would register itself with IGMP.
3. The receiver receives the Path messages.
4. The receiver sends appropriate Resv messages toward the sender. These messages carry a flow descriptor, which is used by routers along the path to make reservations in their link-layer media.
5. The sender receives the Resv message, and then it starts sending application data.

This sequence of events is not necessarily strictly synchronized. For example, receivers can register themselves before receiving Path messages from the sender, and application data can flow before the sender receives Resv messages. Application data that is delivered before the actual reservation contained in the Resv message typically is treated as best effort, nonreal-time traffic with no QoS guarantee.

RSVP Message Types

RSVP uses several types of messages to establish and remove paths for data flows, to establish and remove reservation information, to confirm the establishment of reservations, and to report errors.

Path Messages

Each sender host transmits Path messages downstream along the routes provided by the unicast and multicast routing protocols. Path messages follow the exact paths of application data, creating path states in the routers along the way, thus enabling routers to learn the previous hop and next-hop node for the session. Path messages are sent periodically to refresh path states.

The refresh interval is controlled by a variable called the *refresh time*, which is the periodical refresh timer expressed in seconds. A path state times out if a router does not receive a specified number of consecutive Path messages. This number is specified by a variable called *keep-multiplier*. Path states are kept for $(\text{keep-multiplier} + 0.5) * 1.5 * \text{refresh-time}$ seconds.

Resv Messages

Each receiver host sends reservation request (Resv) messages upstream toward senders and sender applications. Resv messages must follow exactly the reverse path of Path messages. Resv messages create and maintain a reservation state in each router along the way.

Resv messages are sent periodically to refresh reservation states. The refresh interval is controlled by the same refresh time variable, and reservation states are kept for $(keep-multiplier + 0.5) * 1.5 * refresh-time$ seconds.

PathTear Messages

PathTear messages remove (tear down) path states as well as dependent reservation states in any routers along a path. PathTear messages follow the same path as Path messages. A PathTear typically is initiated by a sender application or by a router when its path state times out.

PathTear messages are not required, but they enhance network performance because they release network resources quickly. If PathTear messages are lost or not generated, path states eventually time out when they are not refreshed, and then the resources associated with the path are released.

ResvTear Messages

ResvTear messages remove reservation states along a path. These messages travel upstream toward senders of the session. In a sense, ResvTear messages are the reverse of Resv messages. ResvTear messages typically are initiated by a receiver application or by a router when its reservation state times out.

ResvTear messages are not required, but they enhance network performance because they release network resources quickly. If ResvTear messages are lost or not generated, reservation states eventually time out when they are not refreshed, and then the resources associated with the reservation are released.

PathErr Messages

When path errors occur (usually because of parameter problems in a Path message), the router sends a unicast PathErr message to the sender that issued the Path message. Using PathErr messages is advisory; these messages do not alter any path state along the way.

ResvErr Messages

When a reservation request fails, a ResvErr error message is delivered to all the receivers involved. Using ResvErr messages is advisory; these messages do not alter any reservation state along the way.

ResvConfirm Messages

Receivers can request confirmation of a reservation request, and this confirmation is sent with ResvConfirm message. Because of the complex RSVP flow-merging rules, a confirmation message does not necessarily provide end-to-end confirmation of the entire path. Therefore, ResvConfirm messages are an indication of potential success only, with no guarantees.

The Resource Reservation Protocol (RSVP) is a resource reservation setup protocol that is designed to interact with integrated services on the Internet.

RSVP Reservation Styles

A reservation request includes options for specifying the reservation style. The reservation styles define how reservations for different senders within the same session are treated and how senders are selected.

Two options specify how reservations for different senders within the same session are treated:

- Distinct reservation—Each receiver establishes its own reservation with each upstream sender.
- Shared reservation—All receivers make a single reservation that is shared among many senders.

Two options specify how senders are selected:

- Explicit sender—List all selected senders.
- Wildcard sender—Select all senders, which then participate in the session.

The following reservation styles, formed by a combination of these four options, currently are defined:

- Fixed filter (FF)—This reservation style consists of distinct reservations among explicit senders. Examples of applications that use fixed-filter style reservations are video applications and unicast applications, which both require flows that have a separate reservation for each sender.
- Wildcard filter (WF)—This reservation style consists of shared reservations among wildcard senders. This type of reservation reserves bandwidth for any and all senders, and propagates upstream toward all senders, automatically extending to new senders as they appear. A sample application for wildcard filter reservations is an audio application in which each sender transmits a distinct data stream. Typically, only a few senders are transmitting at any one time. Such a flow does not require a separate reservation for each sender; a single reservation is sufficient.
- Shared explicit (SE)—This reservation style consists of shared reservations among explicit senders. This type of reservation reserves bandwidth for a limited group of senders. A sample application is an audio application similar to that described for wildcard filter reservations.

Chapter 11

RSVP Configuration Guidelines

To configure RSVP, you include statements at the [edit protocols rsvp] hierarchy level of the configuration.

```
protocols {
  rsvp {
    disable;
    keep-multiplier number;
    preemption ( aggressive | disabled | normal );
    refresh-time seconds;
    traceoptions {
      file filename <replace> <size size> <files number> <no-stamp>
        <(world-readable | no-world-readable)>;
      flag flag <flag-modifier> <disable>;
    }
    interface interface-name {
      disable;
      (aggregate | no-aggregate);
      authentication-key key;
      bandwidth bps;
      hello-interval seconds;
      subscription percentage;
    }
  }
}
```

By default, RSVP is disabled.

This chapter describes the minimum required configuration and discusses the following tasks for configuring RSVP:

- Enable RSVP on page 122
- Configure RSVP Aggregation on page 122
- Configure the RSVP Hello Interval on page 123
- Configure RSVP Authentication on page 123
- Reserve Bandwidth on an Interface on page 124
- Configure RSVP Timers on page 124

- Preempt RSVP Sessions on page 125
- Trace RSVP Protocol Traffic on page 125
- Configure RSVP and MPLS on page 127

Minimum RSVP Configuration

To enable RSVP on all interfaces, include the following statement in the configuration file. All other RSVP configuration statements are optional.

```
[edit]
protocols {
  rsvp {
    interface all;
  }
}
```

Enable RSVP

To enable RSVP, including the following statements at the [edit] hierarchy level:

```
[edit]
protocols {
  rsvp {
    interface interface-name;
  }
}
```

To enable RSVP on all interfaces, specify *all* for *interface-name*.

If you have configured interface properties on a group of interfaces and want to disable RSVP on one of the interfaces, include the *disable* statement within the *rsvp* interface statement:

```
[edit]
protocols {
  rsvp {
    interface interface-name {
      disable;
    }
  }
}
```

Configure RSVP Aggregation

The resource requirements—processing, bandwidth, and memory—for running RSVP on a router increase proportionally with the number of sessions. Handling large numbers of refresh messages transmitted between RSVP neighbors is crucial for supporting large numbers of sessions. Path and Resv messages typically represent the majority of refreshes.

If topology failures occur, every node adjacent to the failure might notify all affected sender and receiver nodes. These notification messages are either tear or error messages, and they typically represent a flood that ripples out from the original failure point.

Aggregation provides a mechanism for reducing message flooding and network overload. It also enhances the efficiency and reliability in delivering RSVP tear or error messages. Note that RSVP aggregation is called *Bundle Message* in the internet draft *RSVP Refresh Reduction Extensions*.

By default, aggregation is disabled on all interfaces. For interoperability with other routers, you might need to keep aggregation disabled. However, we recommend that you enable aggregation between Juniper Networks routers to improve scalability. If you have several thousand MPLS LSPs, you must enable aggregation to ensure stable operation. To enable aggregation, include the aggregate statement at the [edit protocols rsvp interface *interface-name*] hierarchy level:

```
[edit protocols rsvp interface interface-name]  
aggregate;
```

Configure the RSVP Hello Interval

RSVP hello packets enable RSVP nodes to detect the loss of a neighboring node's RSVP state information. (Losses typically occur when the neighboring router restarts or the link fails.) In standard RSVP, such detection occurs as a consequence of RSVP's soft-state model. However, detection typically requires several minutes to time out the soft state. RSVP hello packets detect the neighboring node's state changes much more quickly, usually within 10-20 seconds.

Between hello-capable neighbors, hello packets are sent unicast toward each other. A loss of $(2 \times \text{keep-multiplier} + 1)$ consecutive hello packets causes the neighbor's state to go down, and all RSVP sessions to and from that neighbor are declared to be down.

JUNOS RSVP hello packets are optional and are backwards compatible with RSVP implementations that do not support hello packets. For neighbors that do not support hello packets, RSVP uses the soft-state timeout for loss detection.

If all neighboring nodes support hello packets, you can reduce the refresh overhead (by increasing the value set in the refresh-time statement) without adversely affecting the node or link failure detection time. Also, the network can scale to a larger number of sessions because the refresh operations consume less CPU and bandwidth. For information about setting the refresh overhead, see "Configure RSVP Timers" on page 124.

By default, RSVP sends hello packets every 3 seconds. To modify how often RSVP sends hello packets, include the hello-interval statement at the [edit protocols rsvp interface *interface-name*] hierarchy level:

```
[edit protocols rsvp interface interface-name]  
hello-interval seconds;
```

Configure RSVP Authentication

All RSVP protocol exchanges can be authenticated to guarantee that only trusted neighbors participate in setting up reservations. By default, RSVP authentication is disabled.

RSVP authentication uses an HMAC-MD5 message-based digest. This scheme produces a message digest based on a secret authentication key and the message contents. (The message contents also include a sequence number.) The computed digest is transmitted with RSVP messages. Once you have configured authentication, all received and transmitted RSVP messages with all neighbors are authenticated on this interface.

MD5 authentication also provides protection against forgery and message modification. However, it does not provide confidentiality because all messages are sent in clear text, and it does not prevent replay attacks.

By default, authentication is disabled. To enable authentication, configure a key on each interface by including the authentication-key statement at the [edit protocols rsvp interface *interface-name*] hierarchy level:

```
[edit protocols rsvp interface interface-name]  
authentication-key key;
```

Reserve Bandwidth on an Interface

For each interface on which RSVP is enabled, by default, RSVP permits all the interface's bandwidth (100 percent) to be used for RSVP reservations.

Oversubscription on an interface occurs when the aggregate demand of all RSVP sessions is allowed to exceed physical capacity of the link. You can use oversubscription to take advantage of the statistical nature of traffic patterns and to permit higher utilization of links. In particular, you can use oversubscription in places where peak utilizations of traffic do not coincide in time.

Undersubscription on an interface occurs when the total demand of all RSVP sessions is always less than the physical capacity of the link. You can use undersubscription to bound utilization of links and reduce congestion.

You can modify the link bandwidth used for RSVP reservations, either decreasing it below 100 percent or oversubscribing the interface. To do this, include the subscription statement at the [edit protocols rsvp interface *interface-name*] hierarchy level:

```
[edit protocols rsvp interface interface-name]  
subscription percentage;
```

percentage is the percentage of the interface's bandwidth that RSVP allows to be used for reservations. It can be a value from 0 through 65000 percent. If you specify a value greater than 100, you are oversubscribing the interface.

You can use the subscription factor to shut down new RSVP sessions on a per-interface basis. If you set the percentage to 0, no new sessions (including those with zero bandwidth requirements) are permitted on the interface. Existing RSVP sessions are not affected by changing the subscription factor. To clear an existing session, issue the clear rsvp session command.

Configure RSVP Timers

RSVP uses two interrelated timing parameters:

- The refresh time controls the interval between the successive generation of refresh messages. Refresh messages include Path and Resv messages. Refresh messages are sent periodically so that reservation states in neighboring nodes do not time out. Each node chooses a value for the refresh timer independently. Each Path and Resv message carries the refresh timer value, and the receiving node extracts this value from the messages.
- The keep multiplier is a locally configured small integer in the range 1 through 255.

To determine the lifetime of a reservation state, use the following formula:

$$lifetime = (keep-multiplier + 0.5) * 1.5 * refresh-time$$

In the worst case, $(keep-multiplier - 1)$ successive refresh messages must be lost before a reservation state is deleted.

By default, the refresh timer value is 30 seconds. To modify this value, include the refresh-time statement at the [edit protocols rsvp] hierarchy level:

```
[edit protocols rsvp]
refresh-time seconds;
```

The default value of the keep multiplier is 3. To modify this value, include the keep-multiplier statement at the [edit protocols rsvp] hierarchy level:

```
[edit protocols rsvp]
keep-multiplier number;
```

Preempt RSVP Sessions

Whenever bandwidth is insufficient to handle all RSVP sessions, you can control the preemption of RSVP sessions. By default, an RSVP session is preempted only by a new higher-priority session.

To always preempt a session when the bandwidth is insufficient, include the preemption aggressive statement at the [edit protocols rsvp] hierarchy level:

```
[edit protocols rsvp]
preemption aggressive;
```

To disable RSVP session preemption, include the preemption disabled statement at the [edit protocols rsvp] hierarchy level:

```
[edit protocols rsvp]
preemption disabled;
```

To return to the default (that is, preempt a session only for a new higher-priority session), include the preemption normal statement at the [edit protocols rsvp] hierarchy level:

```
[edit protocols rsvp]
preemption normal;
```

Trace RSVP Protocol Traffic

To trace RSVP protocol traffic, you can specify options in the global traceoptions statement at the [edit routing-options] hierarchy level, and you can specify RSVP-specific options by including the traceoptions statement at the [edit protocols rsvp] hierarchy level:

```
[edit protocols rsvp]
traceoptions {
  file filename <replace> <size size> <files number> <no-stamp>
    <(world-readable | no-world-readable)>;
  flag flag <flag-modifier> <disable>;
}
```

You can specify the following RSVP-specific flags in the RSVP traceoptions statement:

- error—Trace all detected error conditions.
- packets—Trace all RSVP messages, including Path, Resv, PathTear, ResvTear, PathErr, ResvErr, and ResvConf messages.
- path—Trace Path messages.
- pathtear—Trace PathTear messages.
- resv—Trace Resv messages.
- resvtear—Trace ResvTear messages.
- state—Trace session state transitions.

For general information about tracing and global tracing options, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Examples: Trace RSVP Protocol Traffic

Trace RSVP Path messages in detail:

```
[edit]
protocols {
  rsvp {
    traceoptions {
      file rsvp size 10m files 5;
      flag path;
    }
  }
}
```

Trace all RSVP messages:

```
[edit]
protocols {
  rsvp {
    traceoptions {
      file rsvp size 10m files 5;
      flag packets;
    }
  }
}
```

Trace all RSVP error conditions:

```
[edit]
protocols {
  rsvp {
    traceoptions {
      file rsvp size 10m files 5;
      flag error;
    }
  }
}
```


Configure RSVP and MPLS

The primary purpose of the JUNOS RSVP software is to support dynamic signaling within LSPs. When you enable both MPLS and RSVP on a router, MPLS becomes a client of RSVP. No additional configuration is required to bind MPLS and RSVP.

You can configure MPLS to set up signaled paths using the `label-switched-path` statement. Each LSP translates into a request for RSVP to initiate an RSVP session. This request is passed through the internal interface between label switching and RSVP. After examining the request information, checking RSVP states and the local routing tables, RSVP initiates one session for each LSP. The session is sourced from the local router and is destined to the target of the LSP.

When an RSVP session is successfully created, the LSP is set up along the paths created by the RSVP session. If the RSVP session is unsuccessful, RSVP notifies MPLS of its status. It is up to MPLS to try to initiate backup paths or to continue retrying the initial path.

To pass label-switching signaling information, RSVP supports four additional objects: Label Request Object, Label Object, Explicit Route Object, and Record Route Object. For an LSP to be set up successfully, all routers along the path must support MPLS, RSVP, and these four objects. Of the four objects, Record Route Object is not mandatory.

To configure MPLS and make it a client of RSVP, do the following:

- Enable MPLS on all routers that are to participate in label switching (that is, on all routers that might be part of a label-switching path).
- Enable RSVP on all routers and on all router interfaces that form the LSP.
- Configure the routers that are to be the beginning of the LSP.

Example: Configure RSVP and MPLS

The following shows a sample configuration for a router at the beginning of an LSP:

```
[edit]
protocols {
  mpls {
    label-switched-path sf-to-london {
      to 192.168.1.4;
    }
  }
  rsvp {
    interface so-0/0/0;
  }
}
```

The following shows a sample configuration for all the other routers that form the LSP:

```
[edit]
protocols {
  mpls {
    interface so-0/0/0;
  }
  rsvp {
    interface so-0/0/0;
  }
}
```

.....

Chapter 12

Summary of RSVP Configuration Statements

This chapter provides a reference for each of the RSVP configuration statements. The statements are organized alphabetically.

aggregate

Syntax	(aggregate no-aggregate);
Hierarchy Level	[edit protocols rsvp interface <i>interface-name</i>]
Description	<p>Control the use of RSVP aggregate messages on an interface:</p> <ul style="list-style-type: none">■ aggregate—Use RSVP aggregate messages.■ no-aggregate—Do not use RSVP aggregate messages. <p>Aggregate messages can pack multiple RSVP messages into a single transmission, thereby reducing network overhead and enhancing efficiency. The number of supportable sessions and processing overhead are significantly improved when aggregation is enabled.</p> <p>Not all routers connected to a subnet need to support aggregation simultaneously. Each RSVP router negotiates its intention to use aggregate messages on a per-neighbor basis. Only when both routers agree are aggregate messages sent.</p>
Default	Aggregation is disabled.
Usage Guidelines	See “Configure RSVP Aggregation” on page 122.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

authentication-key

Syntax	authentication-key <i>key</i> ;
Hierarchy Level	[edit protocols rsvp interface <i>interface-name</i>]
Description	<p>Authentication key (password). Neighboring routers use the password to verify the authenticity of packets sent from this interface.</p> <p>RSVP uses HMAC-MD5 authentication, which is defined in RFC 2104, <i>HMAC: Keyed-Hashing for Message Authentication</i>.</p> <p>All routers that are connected to the same IP subnet must use the same authentication scheme and password.</p>
Options	<i>key</i> —Authentication password. It can be 1 to 16 contiguous digits or letters. Separate decimal digits with periods. Separate hexadecimal digits with periods and precede the string with 0x. If you include spaces in the password, enclose the entire password in quotation marks (" ").
Usage Guidelines	See "Configure RSVP Authentication" on page 123.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

bandwidth

Syntax	bandwidth <i>bps</i> ;
Hierarchy Level	[edit protocols rsvp interface <i>interface-name</i>]
Description	For certain logical interfaces (such as ATM, PVC or frame relay), you cannot determine the correct bandwidth from the hardware. This statement allows you to specify the actual available bandwidth.
Default	The hardware raw bandwidth is used.
Options	<p><i>bps</i>—Bandwidth is specified in bits per second. You can specify this as an integer value (if you do so, count your zeros carefully, or you can use the abbreviations k (for a thousand), m (for a million), or g (for a billion [also called a thousand million]).</p> <p>Range: Any positive integer</p> <p>Default: 0 (no bandwidth is reserved)</p>
Usage Guidelines	See "Reserve Bandwidth on an Interface" on page 124
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

disable

Syntax	disable;
Hierarchy Level	[edit protocols rsvp interface <i>interface-name</i>]
Description	Explicitly disable RSVP on an interface.
Default	RSVP is enabled on interfaces configured with the RSVP interface statement.
Usage Guidelines	See “Enable RSVP” on page 122.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

hello-interval

Syntax	hello-interval <i>seconds</i> ;
Hierarchy Level	[edit protocols rsvp interface <i>interface-name</i>]
Description	Enable the sending of hello packets on the interface. If you configure a nonzero hello interval and (2 x keep-multiplier + 1) consecutive hello exchanges with a neighbor are lost, the neighbor and all sessions to and from that neighbor are declared to be down.
Options	<i>seconds</i> —Length of time between hello packets. A value of 0 disables the sending of hello packets on the interface. Range: 1 through 60 Default: 3 seconds
Usage Guidelines	See “Configure the RSVP Hello Interval” on page 123
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

interface

Syntax	interface <i>interface-name</i> { disable; authentication-key <i>key</i> ; subscription <i>percentage</i> ; }
Hierarchy Level	[edit protocols rsvp]
Description	Enable RSVP on one or more router interfaces.
Default	RSVP is disabled on all interfaces.

Options *interface-name*—Name of an interface. To configure all interfaces, you can specify all. For details about specifying interfaces, see the *JUNOS Internet Software Configuration Guide: Interfaces, Class of Service, and Firewalls*.

The remaining statements are explained separately.

Usage Guidelines See “Enable RSVP” on page 122.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

keep-multiplier

Syntax keep-multiplier *number*;

Hierarchy Level [edit protocols rsvp]

Description Set the keep multiplier value.

Options *number*—Multiplier value.
Range: 1 through 255
Default: 3

Usage Guidelines See “Configure RSVP Timers” on page 124.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-aggregate

See aggregate on page 129

preemption

Syntax preemption (aggressive | disabled | normal);

Hierarchy Level [edit protocols rsvp]

Description Control RSVP session preemption.

Options aggressive—Preempt RSVP sessions whenever bandwidth is insufficient to handle all sessions. A session is preempted whenever bandwidth is lowered or a new higher-priority session is established.

disabled—Do not preempt RSVP sessions.

normal—Preempt RSVP sessions to accommodate new higher-priority sessions, when bandwidth is insufficient to handle all sessions.

Default	normal
Usage Guidelines	See “Preempt RSVP Sessions” on page 125.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

refresh-time

Syntax	refresh-time <i>seconds</i> ;
Hierarchy Level	[edit protocols rsvp]
Description	Set the refresh time.
Options	<i>seconds</i> —Refresh time. Range: 1 through 65535 Default: 30 seconds
Usage Guidelines	See “Configure RSVP Timers” on page 124.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

rsvp

Syntax	rsvp { ... }
Hierarchy Level	[edit protocols]
Description	Enable RSVP routing on the router. You must include the rsvp statement in the configuration to enable RSVP on the router. See “Minimum RSVP Configuration” on page 122.
Default	RSVP is disabled on the router.
Usage Guidelines	See “Enable RSVP” on page 122.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

subscription

Syntax	subscription <i>percentage</i> ;
Hierarchy Level	[edit protocols rsvp interface <i>interface-name</i>]
Description	<p>Configure the subscription factor on the interface, which is the percentage of the link bandwidth that can be used for the RSVP reservation process.</p> <p>You can use the subscription factor to shut down new RSVP sessions on a per-interface basis. If you set the percentage to 0, no new sessions (including those with zero bandwidth requirements) are permitted on the interface. Existing RSVP sessions are not affected by changing the subscription factor. To clear an existing session, issue the clear rsvp session command.</p>
Options	<p><i>percentage</i>—Percentage of the interface's bandwidth that RSVP allows to be used for reservations. If you specify a value greater than 100, you are oversubscribing the interface.</p> <p>Range: 0 through 65000</p> <p>Default: 100 percent</p>
Usage Guidelines	See "Reserve Bandwidth on an Interface" on page 124.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

traceoptions

Syntax	<pre>traceoptions { file <i>filename</i> <replace> <size <i>size</i>> <files <i>number</i>> <no-stamp> <(world-readable no-world-readable)>; flag <i>flag</i> <<i>flag-modifier</i>> <disable>; }</pre>
Hierarchy Level	[edit protocols rsvp]
Description	RSVP protocol-level trace options.
Default	The default RSVP protocol-level trace options are those inherited from the routing protocols traceoptions statement included at the [edit routing-options] hierarchy level.
Options	<p>disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.</p> <p><i>filename</i>—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory /var/log. We recommend that you place RSVP tracing output in the file rsvp-log.</p> <p><i>files number</i>—(Optional) Maximum number of trace files. When a trace file named <i>trace-file</i> reaches its maximum size, it is renamed <i>trace-file.0</i>, then <i>trace-file.1</i>, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.</p> <p>If you specify a maximum number of files, you also must specify a maximum file size with the size option.</p> <p>Range: 2 to 1000.</p> <p>Default: 2 files</p>

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple *flag* statements.

RSVP Tracing Flags

- *error*—All detected error conditions
- *packets*—All RSVP messages, including Path, Resv, PathTear, ResvTear, PathErr, ResvErr, and ResvConf messages
- *path*—Path messages
- *pathtear*—PathTear messages
- *resv*—Resv messages
- *resvtear*—ResvTear messages
- *state*—Session state transitions

Global Tracing Flags

- *all*—All tracing operations
- *general*—A combination of the normal and route trace operations
- *normal*—All normal operations
Default: If you do not specify this option, only unusual or abnormal operations are traced.
- *policy*—Policy operations and actions
- *route*—Routing table changes
- *state*—State transitions
- *task*—Interface transactions and processing
- *timer*—Timer usage

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- *detail*—Provide detailed trace information
- *receive*—Packets being received
- *send*—Packets being transmitted

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Disallow any user to read the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file.0*. When the *trace-file* again reaches its maximum size, *trace-file.0* is renamed *trace-file.1* and *trace-file* is renamed *trace-file.0*. This renaming scheme continues until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the files option.

Syntax: *xk* to specify KB, *xm* to specify MB, or *xg* to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Usage Guidelines See “Trace RSVP Protocol Traffic” on page 125.

Required Privilege Level routing and trace—To view this statement in the configuration
routing-control and trace-control—To add this statement to the configuration

Part 4

LDP

- LDP Overview on page 139
- Configure LDP on page 145
- Summary of LDP Configuration Statements on page 157



Chapter 13

LDP Overview

Label Distribution Protocol (LDP) is a protocol for distributing labels in nontraffic-engineered applications. LDP allows routers to establish label-switched paths (LSPs) through a network by mapping network-layer routing information directly to data link layer-switched paths.

These LSPs might have an end point at a directly attached neighbor (comparable to IP hop-by-hop forwarding), or they might have an end point at a network egress node, enabling switching through all intermediary nodes. LSPs established by LDP can also traverse traffic-engineered LSPs created by RSVP.

This chapter discusses the following topics:

- Overview on page 139
- LDP Standards on page 140
- JUNOS LDP Protocol Implementation on page 140
- LDP Operation on page 140
- LDP Label Filtering on page 140
- Tunneling LDP LSPs in RSVP LSPs on page 141
- LDP Message Types on page 142

Overview

LDP associates a set of destinations (route prefixes and router addresses) with each data link LSP. This set of destinations is called the Forwarding Equivalence Class (FEC). These destinations all share a common data LSP path egress and a common unicast routing path. Each router chooses the label advertised by the next hop for the FEC and splices it to the label it advertises to all other routers. This forms a tree of LSPs that converge on the egress router.

You can implement Virtual Private Networks (VPNs) using MPLS for tunneling. This allows the use of overlapping address spaces by different VPNs. Some of these MPLS-based approaches to VPNs support only LDP for signaling. With JUNOS implementation of LDP, and Juniper Networks routers at the core of a network, you can implement edge devices that support VPNs using LDP signaling for MPLS.

LDP Standards

LDP is described in *Label Distribution Protocol (LDP)—Version 1 Functional Specification*, Internet draft draft-ietf-mpls-ldp-06.txt.

To access Internet RFCs and drafts, go to the IETF web site www.ietf.org.

JUNOS LDP Protocol Implementation

The JUNOS implementation of LDP supports LDP Version 1. The JUNOS software supports a simple mechanism for tunneling between routers in an IGP, to eliminate the required distribution of external routes within the core. JUNOS allows an MPLS tunnel next hop to all egress routers in the network, with only an IGP running in the core to distribute routes to egress routers. Edge routers run BGP but do not distribute external routes to the core. Instead, the recursive route lookup at the edge resolves to an LSP switched to the egress router. No external routes are necessary.

LDP Operation

You must configure LDP for each interface on which you want LDP to run. LDP creates LSP trees rooted at each egress router for the router ID address that is the subsequent BGP next hop. The ingress point is at every router running LDP. This process provides an inet.3 route to every egress router. If BGP is running, it will attempt to resolve next hops by using the inet.3 table first, which binds most, if not all, of the BGP routes to MPLS tunnel next hops.

Two adjacent routers running LDP become neighbors. If the two routers are connected by more than one interface, they become neighbors on each interface. When LDP routers become neighbors, they establish an LDP session to exchange label information. If per-router labels are in use on both routers, only one LDP session is established between them, even if they are neighbors on multiple interfaces. For this reason, an LDP session is not related to a particular interface.

LDP operates in conjunction with a unicast routing protocol. LDP installs LSPs only when both LDP and the routing protocol are enabled. For this reason, you must enable both LDP and the routing protocol on the same set of interfaces. If this is not done, LSPs might not be established between each egress router and all ingress routers, which might result in loss of BGP-routed traffic.

For LDP to run on an interface, MPLS must be enabled on a logical interface on that interface. For more information, see the *JUNOS Internet Software Configuration Guide: Interfaces, Class of Service, and Firewalls*.

LDP Label Filtering

You can apply policy filters to labels received from and distributed to other routers through LDP. Policy filters provide you with a mechanism to control the establishment of LSPs.

Tunneling LDP LSPs in RSVP LSPs

If you are using RSVP for traffic engineering, you can run LDP simultaneously to eliminate the distribution of external routes in the core. The LSPs established by LDP will be tunneled through the LSPs established by RSVP. LDP effectively treats the traffic-engineered LSPs as single hops.

When you configure the router to run LDP across RSVP-established LSPs, LDP will automatically establish sessions with the router at the other end of the LSP. LDP control packets are routed hop-by-hop, rather than carried through the LSP. This allows you to use simplex (one-way) traffic-engineered LSPs. Traffic in the opposite direction flows through LDP-established LSPs that follow unicast routing rather than through traffic-engineered tunnels.

Label Operations

Figure depicts an LDP LSP being tunneled through an RSVP LSP. (For definitions of label operations, see “Label Description” on page 21.) The shaded inner oval represents the RSVP domain, while the outer oval depicts the LDP domain. RSVP establishes an LSP through routers B, C, D, and E, with the sequence of labels L3, L4. LDP establishes an LSP through routers A, B, E, F, and G, with the sequence of labels L1, L2, L5. LDP views the RSVP LSP between routers B and E as a single hop.

When the packet arrives at Router A, it enters the LSP established by LDP, and a label (L1) is pushed onto the packet. When the packet arrives at router B, the label (L1) is swapped with another label (L2). Because the packet is entering the traffic-engineered LSP established by RSVP, a second label (L3) is pushed onto the packet.

This outer label (L3) is swapped with a new label (L4) at the intermediate router (C) within the RSVP LSP tunnel, and when the penultimate router (D) is reached, the top label is popped. Router E swaps the label (L2) with a new label (L5), and the penultimate router for the LDP-established LSP (F) pops the last label.

Figure 17: Swap and Push Label Operation When Tunneling LDP LSPs through RSVP LSPs

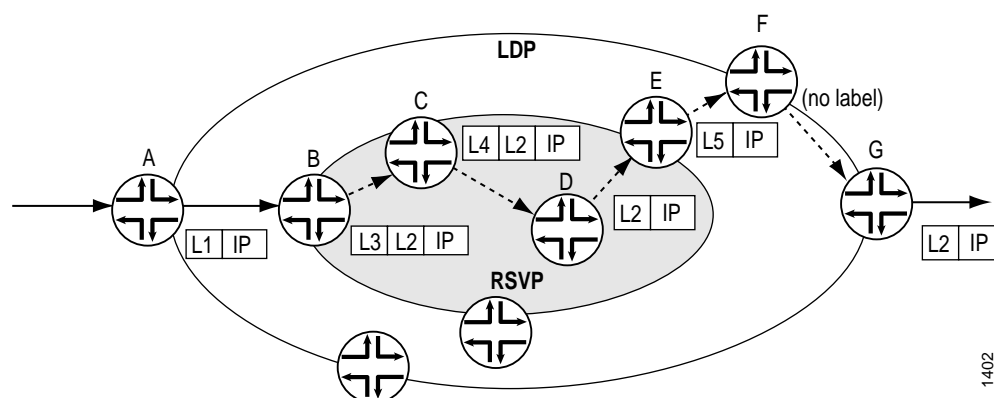
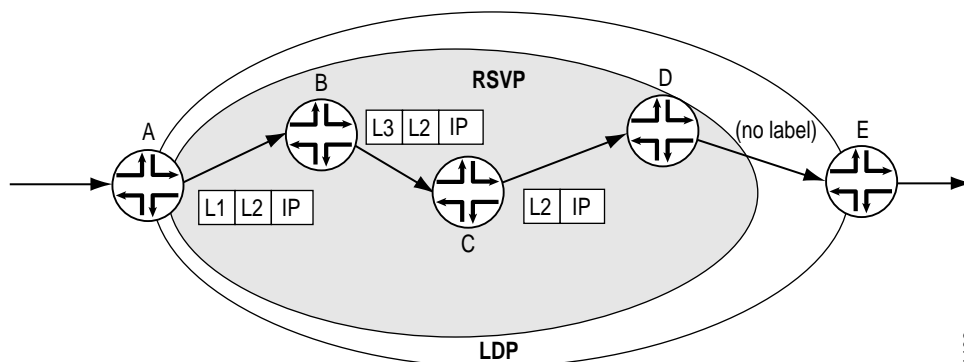


Figure 18 depicts double push label operation (L1L2), which is used when the ingress router (A) of the LDP and the RSVP are the same router. Note that router D is the penultimate hop for the LDP-established LSP, so L2 is popped from the packet by router D.

Figure 18: Double Push Label Operation When Tunneling LDP LSPs through RSVP LSPs



Restrictions for LDP over RSVP

The IGP shortcut computation imposes some restrictions on the network topology allowed. All the routers in the traffic engineered core and in the surrounding LDP cloud must belong to the same OSPF area or IS-IS level. Using multiple areas or levels prevents the IGP shortcut computation from finding an RSVP LSP next hop. As a result, you cannot use a label from a remote LDP session for this router.

If all the routers do not belong to the same area or level, traffic engineering shortcuts must be explicitly enabled in the IGP. (For more information, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.) You cannot use RIP as an IGP for shortcuts.

LDP Message Types

LDP uses several types of messages to establish and remove mappings and to report errors. All LDP messages have a common structure that uses a type-length-value (TLV) encoding scheme.

Discovery Messages

Discovery messages announce and maintain the presence of a router in a network. Routers indicate their presence in a network by sending the hello message periodically. This hello message is transmitted as a UDP packet to the LDP port at the group multicast address for all routers on the subnet.

Session Messages

Session messages establish, maintain, and terminate sessions between LDP peers. When a router establishes a session with another router learned through the hello message, it uses the LDP initialization procedure over TCP transport. When the initialization procedure completes successfully, the two routers are LDP peers, and can exchange advertisement messages.

Advertisement Messages

Advertisement messages create, change, and delete label mappings for Forwarding Equivalence Classes (FECs). Requesting a label or advertising a label mapping to a peer is a decision made by the local router. In general, the router requests a label mapping from a neighboring router when it needs one and advertises a label mapping to a neighboring router when it wants the neighbor to use a label.

Notification Messages

Notification messages provide advisory information and signal error information. LDP sends notification messages to report errors and other events of interest. There are two kinds of LDP notification messages:

- Error notifications signal fatal errors. If a router receives an error notification from a peer for an LDP session, it terminates the LDP session by closing the TCP transport connection for the session and discarding all label mappings learned through the session.
- Advisory notifications pass a router information about the LDP session or the status of some previous message received from the peer.

.....

Chapter 14

Configure LDP

To configure LDP, you include statements at the [edit protocols ldp] hierarchy level of the configuration.

```
[edit protocols]
ldp {
  import [policy-name];
  deaggregate | no-deaggregate;
  egress-policy policy-name;
  export [policy-name];
  keepalive-interval seconds;
  keepalive-timeout seconds;
  preference preference;
  transport-address ( interface | loopback );
  interface interface-name {
    disable;
    hello-interval seconds;
    hold-time seconds;
    deaggregate | no-deaggregate;
    transport-address ( interface | loopback );
  }
  traceoptions {
    file filename <replace> <size size> <files number> <no-stamp>
      <(world-readable | no-world-readable)>;
    flag flag <flag-modifier> <disable>;
  }
}
```

By default, LDP is disabled.

This chapter describes the minimum required configuration and discusses the following tasks for configuring LDP:

- Minimum LDP Configuration on page 146
- Enable LDP on page 146
- Configure the LDP Hello Interval on page 147
- Configure the LDP Hold Time on page 147
- Configure the LDP Keepalive Interval on page 147
- Configure the LDP Keepalive Timeout on page 147
- Configure LDP Route Preferences on page 148

- Configure LDP Received Label Filtering on page 148
- Configure LDP Outbound Label Filtering on page 150
- Enable LDP over RSVP-Established LSPs on page 152
- Configure LDP Transport Address Control on page 152
- Configure LDP Egress Policy on page 152
- Configure FEC Deaggregation on page 153
- Trace LDP Protocol Traffic on page 154

For an LDP configuration example, see “Example: LDP Configuration” on page 155.

Minimum LDP Configuration

To enable LDP on all interfaces, include the following statement in the configuration file. All other LDP configuration statements are optional.

```
[edit]
protocols {
  ldp {
    interface all;
  }
}
```

Enable LDP

To enable LDP on a specific interface, include the following statements at the [edit] hierarchy level:

```
[edit]
protocols {
  ldp {
    interface interface-name;
  }
}
```

To enable LDP on all interfaces, specify *all* for *interface-name*.

If you have configured interface properties on a group of interfaces and want to disable LDP on one of the interfaces, include the disable statement within the LDP interface statement:

```
[edit]
protocols {
  ldp {
    ldp interface-name {
      disable;
    }
  }
}
```

Configure the LDP Hello Interval

LDP hello messages enable LDP nodes to discover one another and to detect the failure of a neighbor or of the link to the neighbor. Hello messages are sent periodically on all interfaces where LDP is enabled.

By default, LDP sends hello messages every 5 seconds. To modify how often LDP sends hello packets, include the hello-interval statement at the [edit protocols ldp interface *interface-name*] hierarchy level:

```
[edit protocols ldp interface interface-name]
hello-interval seconds;
```

Configure the LDP Hold Time

The hold time determines how long an LDP node should wait for a hello message before declaring a neighbor to be down. This value is sent as part of a hello message so that each LDP node tells its neighbors how long to wait. The values sent by each neighbor do not have to match. The hold time should be at least three times the hello interval. The default is 15 seconds. To modify the hold time, include the hold-time statement at the [edit protocols ldp interface *interface-name*] hierarchy level:

```
[edit protocols ldp interface interface-name]
hold-time seconds;
```

Configure the LDP Keepalive Interval

The keepalive interval determines how often a message is sent over the session to ensure that the keepalive timeout is not exceeded. If no other LDP traffic is sent over the session in this much time, a keepalive message is sent. The default is 10 seconds. To modify the keepalive interval, include the keepalive-interval statement at the [edit protocols ldp interface *interface-name*] hierarchy level:

```
[edit protocols ldp interface interface-name]
keepalive-interval seconds;
```

Configure the LDP Keepalive Timeout

After an LDP session is established, messages must be exchanged periodically to ensure that the session is still working. The keepalive timeout defines the amount of time that the neighbor LDP node waits before deciding that the session has failed. This value is usually set to at least three times the keepalive interval. The default is 30 seconds. To modify the keepalive interval, include the keepalive-timeout statement at the [edit protocols ldp interface *interface-name*] hierarchy level:

```
[edit protocols ldp interface interface-name]
keepalive-timeout seconds;
```

Configure LDP Route Preferences

When several protocols calculate routes to the same destination, route preferences are used to select which route is installed in the forwarding table. The route with the lowest preference value is selected. The preference value can be a number in the range 0 through 255. By default, LDP routes have a preference value of 9. To modify the route preferences, include the preference statement at the [edit protocols ldp] hierarchy level:

```
[edit protocols ldp]
  preference preference;
```

Configure LDP Received Label Filtering

You can filter received LDP label bindings, applying policies to accept or deny bindings advertised by neighboring routers. To configure received label filtering, include the import statement at the [edit protocols ldp interface *interface-name*] hierarchy level.

```
[edit protocols ldp]
  import [policy-name];
```

The named policy (configured at the [edit policy-options] hierarchy level) is applied to all label bindings received from all LDP neighbors. All filtering is done using from statements. Table 2 lists the only from operators that apply to LDP received label filtering.

Table 2: from Operators That Apply to LDP Received Label Filtering

from Operator	Description
interface	Matches on bindings received from a neighbor that is adjacent over the specified interface.
neighbor	Matches on bindings received from the specified LDP router ID.
nexthop	Matches on bindings received from a neighbor advertising the specified interface address.
route-filter	Matches on bindings with the specified prefix.

If a binding is filtered, it still appears in the LDP database, but is not considered for installation as part of a label-switched path.

Generally, applying policies in LDP can only be used to block the establishment of LSPs, not to control their routing. This is because the path that an LSP follows is determined by unicast routing, and not by LDP. However, when there are multiple equal-cost paths to the destination through different neighbors, you can use LDP filtering to exclude some of the possible next hops from consideration. (Otherwise, LDP chooses one of the possible next hops at random.)

LDP sessions are not bound to interfaces or interface addresses. LDP advertises only per-router (not per-interface) labels, so if multiple parallel links exist between two routers, only one LDP session is established and it is not bound to a single interface. Be careful, when a router has multiple adjacencies to the same neighbor, to ensure that the filter does what is expected. (Generally, using nexthop and interface is not appropriate in this case.)

If a label has been filtered (meaning that it has been rejected by the policy and is not used to construct an LSP), it is marked as filtered in the database:

```
user@host> show ldp database
Input label database, 10.10.255.1:0-10.10.255.6:0
Label Prefix
3 10.10.255.6/32 (Filtered)
Output label database, 10.10.255.1:0-10.10.255.6:0
Label Prefix
3 10.10.255.1/32 (Filtered)
```

Examples: Configure Received Label Filtering

Accept only /32 prefixes from all neighbors:

```
[edit]
protocols {
  ldp {
    import only-32;
    ...
  }
}
policy-options {
  policy-statement only-32 {
    term first {
      from {
        route-filter 0.0.0.0/0 upto /31;
      }
      then reject;
    }
    then accept;
  }
}
```

Accept 131.108/16 from router ID 10.10.255.2 and accept all prefixes from all other neighbors:

```
[edit]
protocols {
  ldp {
    import nosy-neighbor;
    ...
  }
}
policy-options {
  policy-statement nosy-neighbor {
    term first {
      from {
        neighbor 10.10.255.2;
        route-filter 131.108.0.0/16 orlonger accept;
        route-filter 0.0.0.0/0 orlonger reject;
      }
    }
    then accept;
  }
}
```

Configure LDP Outbound Label Filtering

You can configure export policies to filter LDP outbound labels. You can filter outbound label bindings by applying routing policies to block bindings from being advertised to neighboring routers. To configure outbound label filtering, include the export statement at the [edit protocols ldp] hierarchy level.

```
[edit protocols ldp]
export [policy-name];
```

The named export policy (configured at the [edit policy-options] hierarchy level) is applied to all label bindings transmitted to all LDP neighbors. The only from operator that applies to LDP outbound label filtering is route-filter, which matches bindings with the specified prefix. The only to operators that apply to outbound label filtering are the operators in Table 3:

Table 3: to Operators for LDP Outbound Label Filtering

to Operator	Description
interface	Matches on bindings sent to a neighbor that is adjacent over the specified interface
neighbor	Matches on bindings sent to the specified LDP router ID
nexthop	Matches on bindings sent to a neighbor advertising the specified interface address

If a binding is filtered, the binding is not advertised to the neighboring router, but it can be installed as part of a label-switched path on the local router. You can apply policies in LDP to block the establishment of LSPs, but not to control their routing. The path an LSP follows is determined by unicast routing, not by LDP.

LDP sessions are not bound to interfaces or interface addresses. LDP advertises only per-router (not per-interface) labels. If multiple parallel links exist between two routers, only one LDP session is established and it is not bound to a single interface.

Do not use the nexthop and interface operators when a router has multiple adjacencies to the same neighbor.

Filtered labels are marked in the database:

```
user@host> show ldp database
Input label database, 10.10.255.1:0-10.10.255.3:0
Label Prefix
100007 10.10.255.2/32
3 10.10.255.3/32
Output label database, 10.10.255.1:0-10.10.255.3:0
Label Prefix
3 10.10.255.1/32
100001 10.10.255.6/32 (Filtered)
```


Examples: Configure Outbound Label Filtering

Block transmission of 10.10.255.6/32 to all neighbors:

```
[edit protocols]
ldp {
  export block-one;
}
policy-options {
  policy-statement block-one {
    term first {
      from {
        route-filter 10.10.255.6/32 exact;
      }
      then reject;
    }
    then accept;
  }
}
```

Send only 131.108/16 to router ID 10.10.255.2, and send all prefixes to all other routers:

```
[edit protocols]
ldp {
  export limit-lsps;
}
policy-options {
  policy-statement limit-lsps {
    term allow-one {
      from {
        route-filter 131.108.0.0/16 orlonger;
      }
      to {
        neighbor 10.10.255.2;
      }
      then accept;
    }
    term block-the-rest {
      to {
        neighbor 10.10.255.2;
      }
      then reject;
    }
    then accept;
  }
}
```

Enable LDP over RSVP-Established LSPs

You can run LDP over LSPs established by RSVP, effectively tunneling the LDP-established LSP through the one established by RSVP. To do so, you must enable LDP on the lo0.0 interface (see “Enable LDP” on page 146). Additionally, you must configure the LSPs over which you want LDP to operate, including the `ldp-tunneling` statement:

```
protocols {
  mpls {
    label-switched-path lsp-path-name {
      from source;
      to destination;
      ldp-tunneling;
    }
  }
}
```

For more information on tunneling LDP LSPs, see “Tunneling LDP LSPs in RSVP LSPs” on page 141.

Configure LDP Transport Address Control

You can control the transport address used by LDP. The transport address is the address used for the TCP session over which LDP is running. To configure transport address control, include the `transport-address` statement:

```
transport-address ( loopback | interface );
```

You can configure the transport address globally for all LDP sessions (at the [edit protocols ldp] hierarchy level) or for each interface (at the [edit protocols ldp interface *interface-name*] hierarchy level).

If you select `loopback`, the address of the loopback interface is used as the transport address. If you select `interface`, the interface address is used as the transport address for any LDP sessions to neighbors reachable over that interface.

You cannot use `transport-address interface` when there are multiple parallel links to the same LDP neighbor, because the LDP specification requires that the same transport address be advertised on all interfaces to the same neighbor. If LDP detects multiple parallel links to the same neighbor, it disables interfaces to that neighbor one by one until the condition is cleared, either by disconnecting the neighbor on an interface or by configuring `transport-address loopback`.

Configure LDP Egress Policy

You can control the set of prefixes that are advertised into LDP and cause the router to be the egress router for those prefixes. By default, only the loopback address is advertised into LDP. To configure the set of prefixes from the routing table to be advertised into LDP, include the `egress-policy` statement at the [edit protocols ldp] hierarchy level:

```
[edit protocols ldp]
egress-policy policy-name;
```

The named policy (configured at the [edit policy-options] hierarchy level) is applied to all routes in the routing table. Those routes that match the policy are advertised into LDP. You can control the set of neighbors to which those prefixes are advertised using the export statement. Only from operators are considered; you can use any valid from operator. For more information, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Examples: Configure Egress Policy

Advertise all connected routes into LDP:

```
[edit protocols]
ldp {
    egress-policy connected-only;
}
policy-options {
    policy-statement connected-only {
        from {
            protocol direct;
        }
        then accept;
    }
}
```

Configure FEC Deaggregation

When an LDP egress router advertises multiple prefixes, the prefixes are bound to a single label and aggregated into a single Forwarding Equivalence Class (FEC). By default, LDP maintains this aggregation as the advertisement traverses the network.

By default, because an LSP cannot be split across multiple next hops and all of the prefixes are bound into a single LSP, you cannot load-balance across equal-cost paths.

To change the default to load-balance across equal-cost paths, deaggregate FECs. Deaggregating FECs causes each prefix to be bound to a separate label and become a separate LSP.

To configure deaggregated FECs, include the deaggregate statement:

```
deaggregate;
```

You can configure deaggregated FECs globally for all LDP sessions (at the [edit protocols ldp] hierarchy level) or for each interface (at the [edit protocols ldp interface *interface-name*] hierarchy level).

Deaggregating an FEC allows the resulting multiple LSPs to be distributed across multiple equal-cost paths and distributes LSPs across the multiple next hops on the egress segments but installs only one next hop per LSP.

To aggregate FECs, include the no-deaggregate statement:

```
no-deaggregate;
```

You can configure aggregated FECs globally for all LDP sessions (at the [edit protocols ldp] hierarchy level) or for each interface (at the [edit protocols ldp interface *interface-name*] hierarchy level).

Trace LDP Protocol Traffic

To trace LDP protocol traffic, you can specify options in the global `traceoptions` statement at the `[edit routing-options]` hierarchy level, and you can specify LDP-specific options by including the `traceoptions` statement at the `[edit protocols ldp]` hierarchy level:

```
[edit protocols ldp]
traceoptions {
  file filename <replace> <size size> <files number> <no-stamp>
    <(world-readable | no-world-readable)>;
  flag flag <flag-modifier> <disable>;
}
```

The following trace flags display the operations associated with the sending and receiving of various LDP messages. Each can carry one or more of the following modifiers:

- **address**—Trace the operation of address and address withdrawal messages.
- **binding**—Trace label-binding operations.
- **error**—Trace error conditions.
- **event**—Trace protocol events.
- **initialization**—Trace the operation of initialization messages.
- **label**—Trace the operation of label request, label map, label withdrawal, and label release messages.
- **notification**—Trace the operation of notification messages.
- **packet**—Trace the operation of address, address withdrawal, initialization, label request, label map, label withdrawal, label release, notification, and periodic messages. This modifier is equivalent to setting the address, initialization, label, notification, and periodic modifiers.
- **packet-dump**—Display the contents of the messages selected with the message operation flags.
- **path**—Trace label-switched path operations.
- **periodic**—Trace the operation of hello and keepalive messages.
- **state**—Trace protocol state transitions.

Examples: Trace LDP Protocol Traffic

Trace LDP Path messages in detail:

```
[edit]
protocols {
  ldp {
    traceoptions {
      file ldp size 10m files 5;
      flag path;
    }
  }
}
```

Trace all LDP outgoing messages:

```
[edit]
protocols {
  ldp {
    traceoptions {
      file ldp size 10m files 5;
      flag packets;
    }
  }
}
```

Trace all LDP error conditions:

```
[edit]
protocols {
  ldp {
    traceoptions {
      file ldp size 10m files 5;
      flag error;
    }
  }
}
```

Example: LDP Configuration

The following shows an example of an LDP configuration:

```
[edit]
protocols {
  ldp {
    traceoptions {
      file ldp size 10m files 5 world-readable;
      flag packets receive;
      flag binding;
    }
    interface all {
    }
  }
}
```

.....

Chapter 15

Summary of LDP Configuration Statements

This chapter provides a reference for each of the LDP configuration statements. The statements are organized alphabetically.

deaggregate

Syntax	deaggregate no-deaggregate;
Hierarchy Level	[edit protocols ldp]
Description	Control FEC deaggregation on the router. deaggregate—Deaggregate FECs. no-deaggregate—Aggregate FECs.
Default	Deaggregation is disabled on the router.
Usage Guidelines	See Configure FEC Deaggregation on page 153.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

disable

Syntax	disable;
Hierarchy Level	[edit protocols ldp interface <i>interface-name</i>]
Description	Explicitly disable LDP on an interface.
Default	LDP is enabled on interfaces configured with the LDP interface statement.
Usage Guidelines	See “Enable LDP” on page 146.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

egress-policy

Syntax	egress-policy [<i>policy-name</i>]
Hierarchy Level	[edit protocols ldp]
Description	Control the prefixes advertised into LDP.
Options	<i>policy-name</i> —Name of one or more routing policies.
Default	Only the loopback address is advertised.
Usage Guidelines	See “Configure LDP Egress Policy” on page 152.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

export

Syntax	export [<i>policy-name</i>];
Hierarchy Level	[edit protocols ldp]
Description	Apply policy filters to outbound LDP label bindings. Filters are applied to all label bindings from all neighbors.
Options	<i>policy-name</i> —Name of one or more routing policies.
Usage Guidelines	See “Configure LDP Outbound Label Filtering” on page 150.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

hello-interval

Syntax	hello-interval <i>seconds</i> ;
Hierarchy Level	[edit protocols ldp interface <i>interface-name</i>]
Description	Control the rate at which hello messages are sent on the interface.
Options	<i>seconds</i> —Length of time between hello packets. Range: 1 through 65535 seconds Default: 5 seconds
Usage Guidelines	See “Configure the LDP Hello Interval” on page 147.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

hold-time

Syntax	hold-time <i>seconds</i> ;
Hierarchy Level	[edit protocols ldp interface <i>interface-name</i>]
Description	How long a neighbor should consider the sending router to be operative. The hold time is advertised in LDP hello packets.
Options	<i>seconds</i> —Hold-time value. Range: 1 through 65535 Default: 15 seconds
Usage Guidelines	See “Configure the LDP Hold Time” on page 147.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

import

Syntax	import [<i>policy-name</i>];
Hierarchy Level	[edit protocols ldp]
Description	Apply policy filters to received LDP label bindings. Filters are applied to all label bindings from all neighbors.
Options	<i>policy-name</i> —Name of one or more routing policies.
Usage Guidelines	See “Configure LDP Received Label Filtering” on page 148.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

interface

Syntax	interface <i>interface-name</i> { disable; hello-interval <i>seconds</i> ; hold-time <i>seconds</i> ; deaggregate no-deaggregate; transport-address (interface loopback); }
Hierarchy Level	[edit protocols ldp]
Description	Enable LDP on one or more router interfaces.
Default	LDP is disabled on all interfaces.
Options	<i>interface-name</i> —Name of an interface. To configure all interfaces, you can specify all. The remaining statements are explained separately.

- Usage Guidelines** See “Enable LDP” on page 146.
- Required Privilege Level** routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

keepalive-interval

- Syntax** keepalive-interval *seconds*;
- Hierarchy Level** [edit protocols ldp]
- Description** Set the keepalive interval value.
- Options** *seconds*—Keepalive value.
Range: 1 through 65535
Default: 10 seconds
- Usage Guidelines** See “Configure the LDP Keepalive Interval” on page 147.
- Required Privilege Level** routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

keepalive-timeout

- Syntax** keepalive-timeout *seconds*;
- Hierarchy Level** [edit protocols ldp]
- Description** Set the keepalive timeout value.
- Options** *seconds*—keepalive timeout value.
Range: 1 through 65535
Default: 30 seconds
- Usage Guidelines** See “Configure the LDP Keepalive Timeout” on page 147.
- Required Privilege Level** routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

ldp

- Syntax** ldp { ... }
- Hierarchy Level** [edit protocols]
- Description** Enable LDP routing on the router.

You must include the ldp statement in the configuration to enable LDP on the router.
- Default** LDP is disabled on the router.
- Usage Guidelines** See “Minimum LDP Configuration” on page 146 and “Enable LDP” on page 146.
- Required Privilege Level** routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-deaggregate

See deaggregate on page 157

preference

Syntax	preference <i>preference</i> ;
Hierarchy Level	[edit protocols ldp]
Description	Set the route preference level for LDP routes.
Options	<i>preference</i> —Preferred value. Range: 0 through 255 Default: 9 seconds
Usage Guidelines	See “Configure LDP Route Preferences” on page 148.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

traceoptions

Syntax	<pre> traceoptions { file <i>filename</i> <replace> <size <i>size</i>> <files <i>number</i>> <no-stamp> <(world-readable no-world-readable)>; flag <i>flag</i> <<i>flag-modifier</i>> <disable>; } </pre>
Hierarchy Level	[edit protocols ldp]
Description	LDP protocol-level trace options.
Default	The default LDP protocol-level trace options are those inherited from the routing protocols traceoptions statement included at the [edit routing-options] hierarchy level.
Options	<p>disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.</p> <p><i>filename</i>—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory /var/log. We recommend that you place LDP tracing output in the file ldp-log.</p> <p>files <i>number</i>—(Optional) Maximum number of trace files. When a trace file named <i>trace-file</i> reaches its maximum size, it is renamed <i>trace-file.0</i>, then <i>trace-file.1</i>, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.</p> <p>If you specify a maximum number of files, you also must specify a maximum file size with the size option.</p> <p>Range: 2 to 1000 Default: 2 files</p>

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements.

- address—Operation of address and address withdrawal messages
- binding—Label-binding operations
- error—Error conditions
- event—Protocol events
- initialization—Operation of Initialization messages
- label—Operation of Label Request, Label Map, Label Withdrawal, and Label Release messages
- notification—Operation of Notification messages
- packets—Equivalent to setting address, initialization, label, notification, and periodic
- packet-dump—Contents of the messages selected with the message operation flags.
- periodic—Operation of Hello and Keepalive messages
- path—Label-switched path operations
- state—Protocol state transitions

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- detail—Provide detailed trace information
- receive—Packets being received
- send—Packets being transmitted

no stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Disallow any user to read the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file.0*. When the *trace-file* again reaches its maximum size, *trace-file.0* is renamed *trace-file.1* and *trace-file* is renamed *trace-file.0*. This renaming scheme continues until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the *files* option.

Syntax: *xk* to specify KB, *xm* to specify MB, or *xg* to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Usage Guidelines See “Trace LDP Protocol Traffic” on page 154 and the *JUNOS Internet Software Configuration Guide: Network Management*.

Required Privilege Level routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

transport-address

Syntax transport-address (loopback | interface);

Hierarchy Level [edit protocols ldp],
[edit protocols ldp interface interface-name]

Description Allows control of the transport address used by LDP.

Options loopback—Loopback address is used as the transport address.

interface—First IP address on the interface will be used as the transport address.

Default: loopback

Usage Guidelines See “Configure LDP Transport Address Control” on page 152.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Part 5

CCC

- CCC Overview on page 167
- CCC Configuration on page 169
- Summary of CCC Configuration Statements on page 179

Chapter 16

CCC Overview

Circuit cross-connect (CCC) allows you to configure transparent connections between two circuits, where a circuit can be a Frame Relay DLCI, an ATM VC, a PPP interface, a Cisco HDLC interface, or an MPLS label-switched path (LSP). Using CCC, packets from the source circuit are delivered to the destination circuit with, at most, the Layer 2 address being changed. No other processing—such as header checksums, TTL decrementing, or protocol processing—is done.

CCC circuits fall into two categories: logical interfaces, which include DLCIs, VCs, VLAN IDs, PPP and Cisco HDLC interfaces; and LSPs. The two circuit categories provide three types of cross-connect:

- **Layer 2 switching**—Cross-connects between logical interfaces provide what is essentially Layer 2 switching. The interfaces that you connect must be of the same type.
- **MPLS tunneling**—Cross-connects between interfaces and LSPs allow you to connect two distant interface circuits of the same type by creating MPLS tunnels that use LSPs as the conduit.
- **LSP stitching**—Cross-connects between LSPs provide a way to “stitch” together two label-switched paths, including paths that fall in two different TED areas.

For Layer 2 switching and MPLS tunneling, the cross-connect is bidirectional, so packets received on the first interface are transmitted out the second interface, and those received on the second interface are transmitted out the first. For LSP stitching, the cross-connect is unidirectional.

For CCC connections that connect interfaces, the interfaces must be of the same type; that is, ATM to ATM, Frame Relay to Frame Relay, PPP to PPP, or Cisco HDLC to Cisco HDLC.

Chapter 17

CCC Configuration

This chapter discusses the following cross-connect configuration tasks:

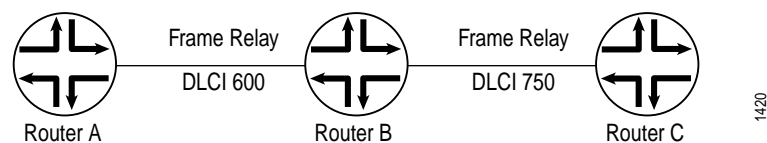
- Configure Layer 2 Switching Cross-Connects on page 169
- Configure MPLS LSP Tunnel Cross-Connects on page 173
- Configure LSP Stitching Cross-Connects on page 177

Configure Layer 2 Switching Cross-Connects

Layer 2 switching cross-connects join logical interfaces to form what is essentially Layer 2 switching. The interfaces that you connect must be of the same type.

Figure 19 illustrates a Layer 2 switching cross-connect. In this topology, Router A and Router C have Frame Relay connections to Router B, which is a Juniper Networks router. CCC allows you to configure Router B to act as a Frame Relay (Layer 2) switch. To do this, you configure a circuit from Router A to Router C that passes through Router B, effectively configuring Router B as a Frame Relay switch with respect to these routers. This configuration allows Router B to transparently switch packets (frames) between Router A and Router C without regard to the packets' contents or the Layer 3 protocols. The only processing that Router B performs is to translate DLCI 600 to 750.

Figure 19: Layer 2 Switching Cross-Connect



If the Router A-to-Router B and Router B-to-Router C circuits were PPP, for example, the Link Control Protocol and Network Control Protocol exchanges occur between Router A and Router C. These messages are handled transparently by Router B, allowing Router A and Router C to use various PPP options (such as header or address compression and authentication) that Router B might not support. Similarly, Router A and Router C exchange keepalives, providing circuit-to-circuit connectivity status.

You can configure Layer 2 switching cross-connects on PPP, Cisco HDLC, Frame Relay, and ATM circuits. In a single cross-connect, only like interfaces can be connected.

To configure Layer 2 switching cross-connects, you must configure the following on the router that is acting as the switch (Router B in Figure 19):

- Define the CCC Encapsulation for Layer 2 Switching Cross-Connects on page 170
- Define the CCC Connection for Layer 2 Switching Cross-Connects on page 171
- Configure MPLS on page 171

Define the CCC Encapsulation for Layer 2 Switching Cross-Connects

To configure Layer 2 switching cross-connects, configure the CCC encapsulation on the router that is acting as the switch (Router B in Figure 19).



You cannot configure families on CCC interfaces; that is, you cannot include the family statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level.

Note

For PPP or Cisco HDLC circuits, specify the encapsulation in the encapsulation statement. This statement configures the entire physical device. For these circuits to work, you must configure a logical interface unit 0.

```
[edit]
interfaces {
  type-fpc/pic/port {
    encapsulation (ppp-ccc | cisco-hdlc-ccc);
    unit 0;
  }
}
```

For ATM circuits, specify the encapsulation when configuring the Virtual Circuit (VC). For each VC, you configure whether it is a circuit or a regular logical interface.

```
[edit]
interfaces {
  at-fpc/pic/port {
    atm-options {
      vpi vpi-identifier maximum-vcs maximum-vcs;
    }
    unit logical-unit-number {
      point-to-point;      # Default interface type
      encapsulation atm-ccc-vc-mux;
      vci vpi-identifier.vci-identifier;
    }
  }
}
```

For Frame Relay circuits, specify the encapsulation when configuring the DLCI. For each DLCI, you configure whether it is a circuit or a regular logical interface. The DLCI for regular interfaces must be in the range 1 through 511. For CCC interfaces, it must be in the range 512 through 1022.

```
[edit]
interfaces {
  interface-switch frame-relay-ccc;
  type-fpc/pic/port {
    unit logical-unit-number {
      point-to-point;      # Default interface type
      encapsulation frame-relay-ccc;
      dlci dlci-identifier;
    }
  }
}
```

Define the CCC Connection for Layer 2 Switching Cross-Connects

To configure Layer 2 switching cross-connects, define the connection between the two circuits. You configure this on the router that is acting as the switch (Router B in Figure 19). The connection joins the interface that comes from the circuit's source to the interface that leads to the circuit's destination. When you specify the interface names, include the logical portion of the name, which corresponds to the logical unit number. The cross-connect is bidirectional, so packets received on the first interface are transmitted out the second interface, and those received on the second interface are transmitted out the first.

```
[edit]
protocols {
  connections {
    interface-switch connection-name {
      interface interface-name.unit-number;
      interface interface-name.unit-number;
    }
  }
}
```

Configure MPLS

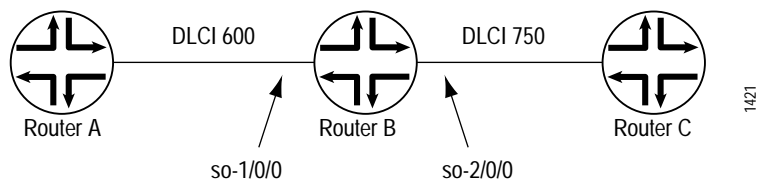
For Layer 2 switching cross-connects to work, you must configure MPLS. The following is a minimal MPLS configuration:

```
[edit]
protocols {
  mpls {
    interface (interface-name | all);
  }
}
```

Example: Configure Layer 2 Switching Cross-Connects

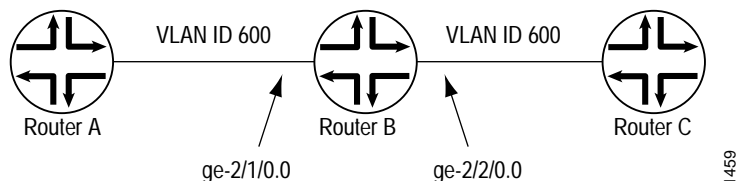
Configure a full-duplex Layer 2 switching cross-connect between Router A and Router C, using a Juniper router, Router B, as the virtual switch. See the topology in Figure 20 and Figure 21.

Figure 20: Sample Topology of Frame Relay Layer 2 Switching Cross-Connect



```
[edit]
interfaces {
  so-1/0/0 {
    encapsulation frame-relay-ccc;
    unit 1 {
      point-to-point;
      encapsulation frame-relay-ccc;
      dlci 600;
    }
  }
  so-2/0/0 {
    encapsulation frame-relay-ccc;
    unit 2 {
      point-to-point;
      encapsulation frame-relay-ccc;
      dlci 750;
    }
  }
}
protocols {
  connections {
    interface-switch router-a-router-c {
      interface so-1/0/0.1;
      interface so-2/0/0.2;
    }
  }
  mpls {
    interface all;
  }
}
```

Figure 21: Sample Topology of a VLAN Layer 2 Switching Cross-Connect

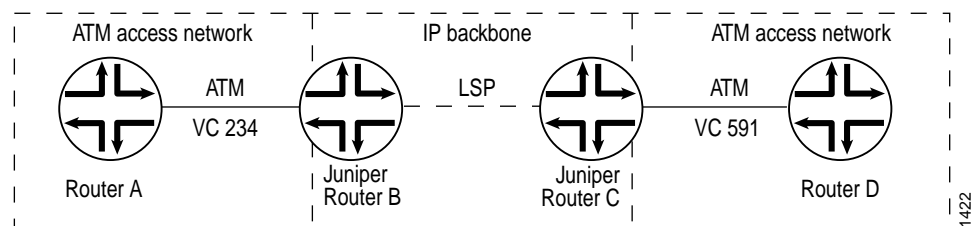


```
[edit]
interfaces {
  ge-2/1/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    unit 0 {
      encapsulation vlan-ccc;
      vlan-id 600;
    }
  }
  ge-2/2/0 {
    vlan-tagging;
    encapsulation vlan-ccc;
    unit 0 {
      encapsulation vlan-ccc;
      vlan-id 600;
    }
    unit 1 {
      family inet {
        vlan-id 1;
        address 10.9.200.1/24;
      }
    }
  }
}
protocols {
  mpls {
    interface all;
  }
  connections {
    interface-switch layer2-sw {
      interface ge-2/1/0.0;
      interface ge-2/2/0.0;
    }
  }
}
```

Configure MPLS LSP Tunnel Cross-Connects

MPLS tunnel cross-connects between interfaces and LSPs allow you to connect two distant interface circuits of the same type by creating MPLS tunnels that use LSPs as the conduit. The topology in Figure 22 illustrates an MPLS LSP tunnel cross-connect. In this topology, two separate networks, in this case ATM access networks, are connected through an IP backbone. CCC allows you to establish an LSP tunnel between the two domains. With LSP tunneling, you tunnel the ATM traffic from one network across a SONET backbone to the second network using an MPLS LSP.

Figure 22: MPLS LSP Tunnel Cross-Connect



When traffic from Router A (VC 234) reaches Router B, it is encapsulated and placed into an LSP, which is sent through the backbone to Router C. At Router C, the label is removed and the packets are placed onto the ATM PVC (VC 591) and sent to Router D. Similarly, traffic from Router D (VC 591) is sent over an LSP to Router B, then placed on VC 234 to Router A.

You can configure LSP tunnel cross-connects on PPP, Cisco HDLC, Frame Relay, and ATM circuits. In a single cross-connect, only like interfaces can be connected.

To configure LSP tunnel cross-connects, you must configure the following on the interdomain router (Router B in Figure 24):

- Define the CCC Encapsulation for LSP Tunnel Cross-Connects on page 174
- Define the CCC Connection for LSP Tunnel Cross-Connects on page 175

When you use MPLS tunnel cross-connects, if you use the default MTU size, IS-IS does not form adjacencies across the tunnel. For the tunnel cross-connects to work, the MTU size on the edge routers (Routers A and D in Figure 22) must be smaller than the LSP's MTU. Use the following calculation to determine the maximum IS-IS MTU size:

$$\text{IS-IS MTU} \leq \text{MPLS MTU} - 4 \text{ bytes} - \text{link-layer overhead}$$

The link-layer overheads varies, depending on the encapsulation:

- ATM—8 bytes
- Frame Relay—2 bytes
- HDLC—4 bytes
- PPP—4 bytes
- VLAN—4 bytes

We recommend that you simply set the MTU to 1497 bytes, which is small enough so that IS-IS works properly.

To modify the MTU, include the `mtu` statement when configuring the logical interface family, at the [edit interfaces *interface-name* unit *logical-unit-number* encapsulation *family*] hierarchy level. For more information about setting the MTU, see the *JUNOS Internet Software Configuration Guide: Interfaces, Class of Service, Firewalls*.

Define the CCC Encapsulation for LSP Tunnel Cross-Connects

To configure LSP tunnel cross-connects, you must configure the CCC encapsulation on the ingress and egress routers (Router B and Router C, respectively, in Figure 22).



Note

You cannot configure families on CCC interfaces; that is, you cannot include the `family` statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level.

For PPP or Cisco HDLC circuits, specify the encapsulation in the encapsulation statement. This statement configures the entire physical device. For these circuits to work, you must configure a logical interface unit 0.

```
[edit]
interfaces {
  type-fpc/pic/port {
    encapsulation (ppp-ccc | cisco-hdlc-ccc);
    unit 0;
  }
}
```

For ATM circuits, specify the encapsulation when configuring the VC. For each VC, you configure whether it is a circuit or a regular logical interface.

```
[edit]
interfaces {
  at-fpc/pic/port {
    atm-options {
      vpi vpi-identifier maximum-vcs maximum-vcs;
    }
    unit logical-unit-number {
      point-to-point;      # Default interface type
      encapsulation atm-ccc-vc-mux;
      vci vpi-identifier.vci-identifier;
    }
  }
}
```

For Frame Relay circuits, specify the encapsulation when configuring the DLCI. For each DLCI, you configure whether it is a circuit or a regular logical interface. The DLCI for regular interfaces must be in the range 1 through 511. For CCC interfaces, it must be in the range 512 through 1022.

```
[edit]
interfaces {
  interface-switch frame-relay-ccc;
  type-fpc/pic/port {
    unit logical-unit-number {
      point-to-point; # default interface type
      encapsulation frame-relay-ccc;
      dlci dlci-identifier;
    }
  }
}
```

For more information about the encapsulation statement, see the *JUNOS Internet Software Configuration Guide: Interfaces and Chassis*.

Define the CCC Connection for LSP Tunnel Cross-Connects

To configure LSP tunnel cross-connects, define the connection between the two circuits on the ingress and egress routers (Router B and Router C, respectively, in Figure 22). The connection joins the interface or LSP that comes from the circuit's source to the interface or LSP that leads to the circuit's destination. When you specify the interface name, include the logical portion of the name, which corresponds to the logical unit number. For the cross-connect to be bidirectional, you must configure cross-connects on two routers.

```
[edit]
```

```

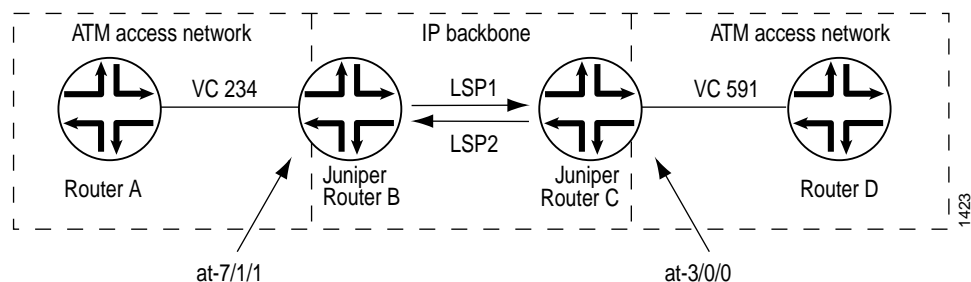
protocols {
  connections {
    remote-interface-switch connection-name {
      interface interface-name.unit-number;
      transmit-lsp label-switched-path;
      receive-lsp label-switched-path;
    }
  }
}

```

Example: Configure LSP Tunnel Cross-Connects

Configure a full-duplex MPLS LSP tunnel cross-connect from Router A to Router D, passing through Router B and Router C. See the topology in Figure 23.

Figure 23: Example Topology of MPLS LSP Tunnel Cross-Connect



On Router B:

```

[edit]
interfaces {
  at-7/1/1 {
    atm-options {
      vpi 1 maximum-vcs 600;
    }
    unit 1 {
      point-to-point; # default interface type
      encapsulation atm-ccc-vc-mux;
      vci 1.234;
    }
  }
}

```

```

protocols {
  connections {
    remote-interface-switch router-b-to-router-c {
      interface at-7/1/1.1;
      transmit-lsp lsp1;
      receive-lsp lsp2;
    }
  }
}

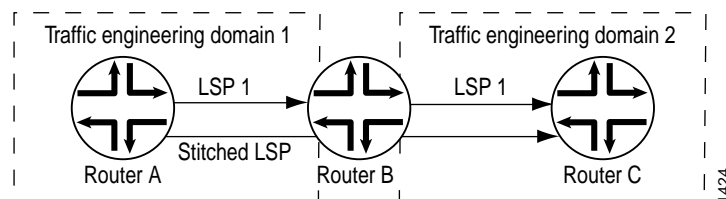
On Router C:
[edit]
interfaces {
  at-3/0/0 {
    atm-options {
      vpi 2 maximum-vc 600;
    }
    unit 2 {
      point-to-point; # default interface type
      interface-switch atm-ccc-vc-mux;
      vci 2.591;
    }
  }
}
protocols {
  connections {
    remote-interface-switch router-b-to-router-c {
      interface at-3/0/0.1;
      transmit-lsp lsp2;
      receive-lsp lsp1;
    }
  }
}

```

Configure LSP Stitching Cross-Connects

LSP stitching cross-connects “stitch” together LSPs to join two LSPs. For example, they stitch together LSPs that fall in two different TED areas. The topology in Figure 24 illustrates an LSP stitching cross-connect. In this topology, the network is divided into two traffic engineering domains. CCC allows you to establish an LSP between the two domains by stitching together LSPs from the two domains. For LSP stitching to work, the LSPs must be dynamic LSPs, not static.

Figure 24: LSP Stitching Cross-Connect



Without LSP stitching, a packet travelling from Router A to Router C is encapsulated on Router A (the ingress router for the first LSP), decapsulated on Router B (the egress router), and then re-encapsulated on Router B (the ingress router for the second LSP). With LSP stitching, you connect LSP1 and LSP2 into a single, stitched LSP, which means that the packet is encapsulated once (on Router A) and decapsulated once (on Router C).

You can use LSP stitching to create a seamless LSP for LSPs carrying any kind of traffic.

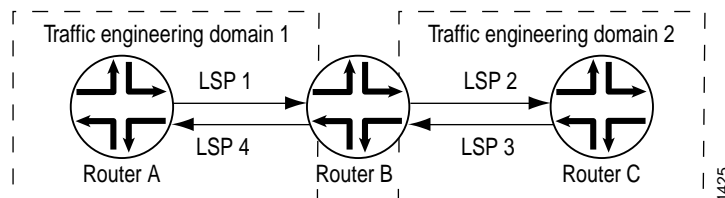
To configure LSP stitching cross-connects, you configure the two LSPs that you are stitching together on the two ingress routers. Then, on the interdomain router (Router B in Figure 24), you define the connection between the two LSPs. The connection joins the LSP that comes from the connection's source to the LSP that leads to the connection's destination.

```
[edit]
protocols {
  connections {
    lsp-switch connection-name {
      transmit-lsp label-switched-path;
      receive-lsp label-switched-path;
    }
  }
}
```

Example: Configure LSP Stitching Cross-Connects

Configure a full-duplex LSP stitching cross-connect between Router A and Router C. To do this, you configure Router B, which is the interdomain router. See the topology in Figure 25.

Figure 25: Example Topology of LSP Stitching Cross-Connect



```
[edit]
protocols {
  connections {
    lsp-switch router-a-to-router-c {
      transmit-lsp lsp2;
      receive-lsp lsp1;
    }
  }
  connections {
    lsp-switch router-c-to-router-a {
      receive-lsp lsp3;
      transmit-lsp lsp4;
    }
  }
}
```

Chapter 18

Summary of CCC Configuration Statements

This chapter provides a reference for each of the circuit cross-connect (CCC) configuration statements. The statements are organized alphabetically.

connections

Syntax	<pre>connections { interface-switch <i>connection-name</i> { interface <i>interface-name.unit-number</i>; interface <i>interface-name.unit-number</i>; } lsp-switch <i>connection-name</i> { transmit-lsp <i>label-switched-path</i>; receive-lsp <i>label-switched-path</i>; } remote-interface-switch <i>connection-name</i> { interface <i>interface-name.unit-number</i>; transmit-lsp <i>label-switched-path</i>; receive-lsp <i>label-switched-path</i>; } }</pre>
Hierarchy Level	[edit protocols]
Description	Define the connection between two circuits in a CCC connection.
Options	The statements are explained separately.
Usage Guidelines	See “CCC Overview” on page 167 and also the <i>JUNOS Internet Software Configuration Guide: Interfaces and Chassis</i> .
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

interface-switch

Syntax interface-switch *connection-name* {
 interface *interface-name.unit-number*;
 interface *interface-name.unit-number*;
 }

Hierarchy Level [edit protocols connections]

Description Configure Layer 2 switching cross-connects. The cross-connect is bidirectional, so packets received on the first interface are transmitted out the second interface, and those received on the second interface are transmitted out the first.

For Layer 2 switching cross-connects to work, you must also configure MPLS.

Options *connection-name*—Connection name.

interface interface-name.unit-number—Interface name. Include the logical portion of the name, which corresponds to the logical unit number.

Usage Guidelines See “Configure Layer 2 Switching Cross-Connects” on page 169.

Required Privilege Level routing—To view this statement in the configuration.
 routing-control—To add this statement to the configuration

lsp-switch

Syntax lsp-switch *connection-name* {
 transmit-lsp *label-switched-path*;
 receive-lsp *label-switched-path*;
 }

Hierarchy Level [edit protocols connections]

Description Configure Layer 2 switching cross-connects.

Options *connection-name*—Connection name.

receive-lsp label-switched-path—Name of the LSP from the connection’s source.

transmit-lsp label-switched-path—Name of the LSP to the connection’s destination.

Usage Guidelines See “CCC Overview” on page 167 and “Configure LSP Stitching Cross-Connects” on page 177.

Required Privilege Level routing—To view this statement in the configuration.
 routing-control—To add this statement to the configuration.

remote-interface-switch

Syntax	remote-interface-switch <i>connection-name</i> { interface <i>interface-name.unit-number</i> ; transmit-lsp <i>label-switched-path</i> ; receive-lsp <i>label-switched-path</i> ; }
Hierarchy Level	[edit protocols connections]
Description	Configure MPLS LSP tunnel cross-connects.
Options	<p><i>connection-name</i>—Connection name.</p> <p>interface <i>interface-name.unit-number</i>—Interface name. Include the logical portion of the name, which corresponds to the logical unit number.</p> <p>receive-lsp <i>label-switched-path</i>—Name of the LSP from the connection's source.</p> <p>transmit-lsp <i>label-switched-path</i>—Name of the LSP to the connection's destination.</p>
Usage Guidelines	See "CCC Overview" on page 167 and "Configure MPLS LSP Tunnel Cross-Connects" on page 173.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

Part 6

VPNs

- Layer 3 VPN Overview on page 185
- Layer 3 VPN Configuration Guidelines on page 201
- Layer 3 VPN Configuration Troubleshooting Guidelines on page 215
- Layer 3 VPN Configuration Examples on page 227
- Summary of Layer 3 VPN Configuration Statements on page 279
- Layer 2 VPN Overview on page 281
- Layer 2 VPN Configuration Guidelines on page 283
- Layer 2 VPN Configuration Example on page 295
- Summary of Layer 2 VPN Configuration Statements on page 311

Chapter 19

Layer 3 VPN Overview

The JUNOS software implements Layer 3 BGP/MPLS Virtual Private Networks (VPNs) as defined in RFC 2547 and Internet draft draft-rosen-rfc2547bis (also referred to as RFC 2547bis). This chapter discusses the following topics that provide background information about Layer 3 VPNs:

- Layer 3 VPN Overview on page 185
- Layer 3 VPN Standards on page 186
- VPN Terminology on page 186
- VPN Attributes on page 187
- VPN-IPv4 Addresses and Route Distinguishers on page 188
- VPN Routing and Forwarding Tables on page 191
- Route Distribution within a VPN on page 194
- Forwarding across the Provider's Core Network on page 198
- Routing Instances for VPNs on page 200

Layer 3 VPN Overview

RFC 2547bis defines a mechanism by which service providers can use their IP backbones to provide VPN services to their customers. A VPN is a set of sites that share common routing information and whose connectivity is controlled by a collection of policies. The sites that make up a VPN are connected over a provider's existing public Internet backbone.

RFC 2547bis VPNs are also known as BGP/MPLS VPNs because BGP is used to distribute VPN routing information across the provider's backbone, and MPLS is used to forward VPN traffic across the backbone to remote VPN sites.

A VPN consists of two topological areas, the provider's network and the customer's network. The provider's network, which runs across the public Internet infrastructure, consists of routers that provide VPN services to a customer's network as well as routers that provide other services. The customer's network is commonly located at multiple physical sites. The provider's network acts to connect the various customer sites in what appears to the customer and the provider to be a private network.

To ensure that VPNs remain private and isolated from other VPNs and from the public Internet, the provider's network maintains policies that keep routing information from different VPNs separate.

A provider can service multiple VPNs as long as its policies keep routes from different VPNs separate. Similarly, a site can belong to multiple VPNs as long as it keeps routes from the different VPNs separate.

Customer networks, because they are private, can use either public addresses or private addresses, as defined in RFC 1918. When customer networks that use private addresses connect to the public Internet infrastructure, the private addresses might overlap with the same private addresses used by other network users. MPLS/BGP VPNs solve this problem by prefixing a VPN identifier to each address from a particular VPN site, thereby creating an address that is unique both within the VPN and within the public Internet. In addition, each VPN has its own VPN-specific routing table that contains the routing information for that VPN only.

Layer 3 VPN Standards

Layer 3 VPNs are defined in the following documents:

- RFC 2547, *BGP/MPLS VPNs*
- Internet draft draft-rosen-rfc2547bis, *BGP/MPLS VPNs*
- RFC 2283, *Multiprotocol Extensions for BGP4*

To access Internet RFCs and drafts, go to the IETF Web site at <http://www.ietf.org>.

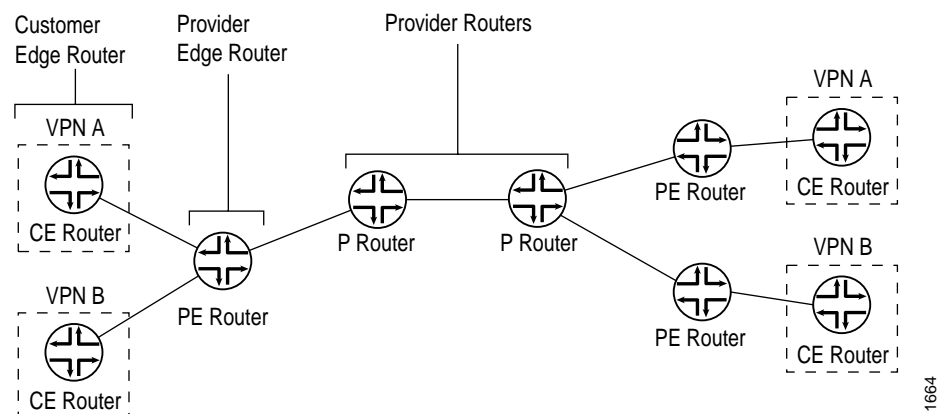
VPN Terminology

VPNs contain the following types of network devices (see Figure 26):

- Provider edge (PE) routers—Routers in the provider's network that connect to CE devices located at customer sites. PE routers support VPN and label functionality. (The label functionality can be provided either by RSVP or LDP.) Within a single VPN, pairs of PE routers are connected through a tunnel, which can be either an MPLS LSP or an LDP tunnel.
- Provider (P) routers—Routers within the core of the provider's network that are not connected to any routers at a customer site but that are part of the tunnel between pairs of PE routers. Provider routers support MPLS LSP or LDP functionality, but do not need to support VPN functionality.
- Customer edge (CE) devices—Routers or switches located at the customer's site that connect to the provider's network. CE devices are typically IP routers.

VPN functionality is provided by the PE routers; the provider and CE routers have no special configuration requirements for VPNs.

Figure 26: VPN Router Components

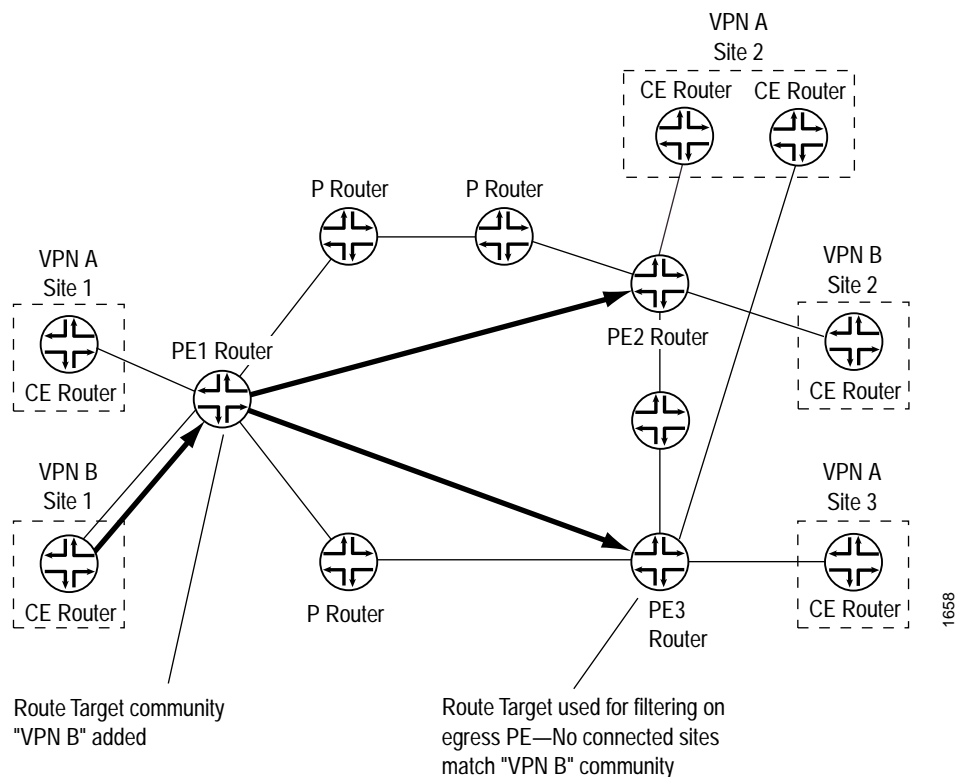


VPN Attributes

Route distribution within a VPN is controlled using BGP extended community attributes. RFC 2547 defines the following three attributes used by VPNs:

- **Target VPN**—Identifies a set of sites within a VPN to which a PE router distributes routes. This attribute is also called the *route target*. The route target is used by the egress PE router to determine whether a received route is destined for a VPN that the router services.
- Figure 27 illustrates the function of the route target. PE Router PE1 adds the route target “VPN B” to routes received from the CE router at Site 1 in VPN B. When it receives the route, the egress router PE2 examines the route target, determines that the route is for a VPN that it services, and accepts the route. When the egress router PE3 receives the same route, it does not accept the route because it does not service any CE routers in VPN B.
- **VPN of origin**—Identifies a set of sites and the corresponding route as having come from one of the sites in that set.
 - **Site of origin**—Uniquely identifies the set of routes that a PE router learned from a particular site. This attribute ensures that a route learned from a particular site through a particular PE-CE connection is not distributed back to the site through a different PE-CE connection. It is particularly useful if you are using BGP as the routing protocol between the PE and CE routers and if different sites in the VPN have not been assigned distinct AS numbers.

Figure 27: VPN Attributes and Route Distribution

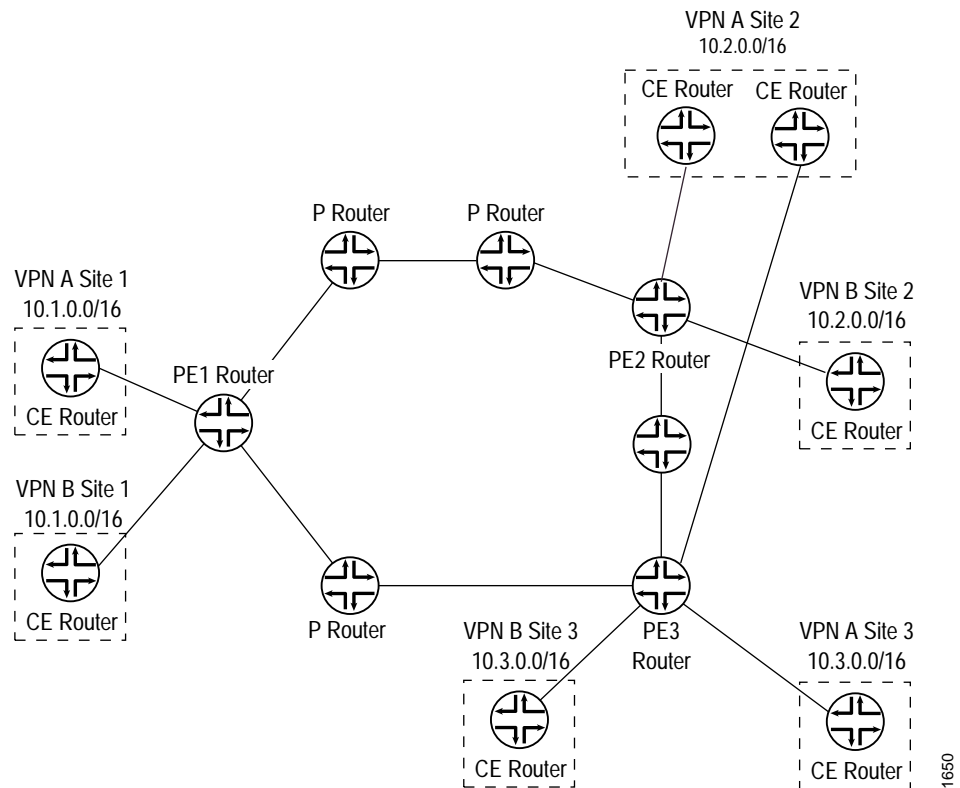


VPN-IPv4 Addresses and Route Distinguishers

Because Layer 3 VPNs connect private networks—which can use either public addresses or private addresses, as defined in RFC 1918—over the public Internet infrastructure, when the private networks use private addresses, the addresses might overlap with the addresses of another private network.

Figure 28 illustrates how private addresses of different private networks can overlap. Here, sites within VPN A and VPN B use the address spaces 10.1.0.0/16, 10.2.0.0/16, and 10.3.0.0/16 for their private networks.

Figure 28: Overlapping Addresses among Different VPNs



To avoid overlapping private addresses, you can configure the network devices to use public addresses instead of private addresses. However, this is a large and complex undertaking. The solution provided in RFC 2547bis uses the existing private network numbers to create a new address that is unambiguous. The new address is part of the VPN-IPv4 address family, which is a BGP address family added as an extension to the BGP protocol. In VPN-IPv4 addresses, a value that identifies the VPN, called a route distinguisher, is prefixed to the private IPv4 address, providing an address that uniquely identifies a private IPv4 address.

Only the PE routers need to support the VPN-IPv4 address extension to BGP. When an ingress PE router receives an IPv4 route from a device within a VPN, it converts it into a VPN-IPv4 route by prefixing the route distinguisher to the route. The VPN-IPv4 addresses are used only for routes exchanged between PE routers. When an egress PE router receives a VPN-IPv4 route, it converts it back to an IPv4 route, by removing the route distinguisher, before announcing the route to its connected CE routers.

VPN-IPv4 addresses have the following format (see Figure 29):

- Route distinguisher—8-byte value that identifies the VPN. The route distinguisher consists of the following fields:
 - Type field (2 bytes)—Determines the length of the other two fields.

If the value in the Type field is 0, the administrator (Adm) field is 4 bytes and the assigned number (AN) field is 2 bytes.

If the value in the Type field is 1, the administrator (Adm) field is 2 bytes and the assigned number (AN) field is 4 bytes.

- Administrator field—Identifies an assigned number authority.

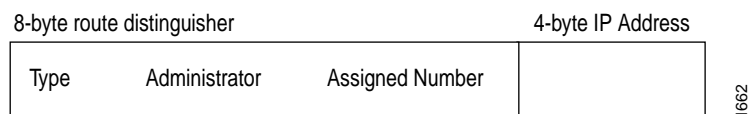
For a Type field value of 0, the administrator field contains an IPv4 address. RFC 2547bis recommends that you use the router's IP address (the address you configure in the router-id statement), which is a nonprivate address.

For a Type field value of 1, the administrator field contains an AS number. RFC 2547bis recommends that you use an IANA-assigned, nonprivate AS number, preferably the ISP's or customer's own AS number.

- Assigned number field—Number assigned by the service provider. For a Type field value of 0, the assigned number field is 2 bytes long. For a Type field value of 1, the assigned number field is 4 bytes.

- IPv4 address—4-byte address of a device within the VPN.

Figure 29: VPN-IPv4 Address Format



To illustrate how the AS number can be used in the route distinguisher, referring to Figure 28, suppose that VPN A is in AS 65535 and VPN B is in AS 666 (both these AS numbers are those of the ISP), and suppose that the route distinguisher for Site 2 in VPN A is 65535:02 and the route distinguisher for Site 2 in VPN B is 666:01. When Router PE2 receives a route from the CE router in VPN A, it converts it from its IP address of 10.2.0.0 to a VPN-IPv4 address of 65535:02:10.2.0.0. When the PE router receives a route from VPN B, which uses the same address space as VPN A, it converts it to a VPN-IPv4 address of 666:02:10.2.0.0.

If the IP address is used in the route distinguisher, suppose the Router PE2's IP address is 172.168.0.1. When the PE router receives a route from VPN A, it converts it to a VPN-IPv4 address of 172.168.0.1:0:10.2.0.0/16, and it converts a route from VPN B to 172.168.0.0:1:10.2.0.0/16.

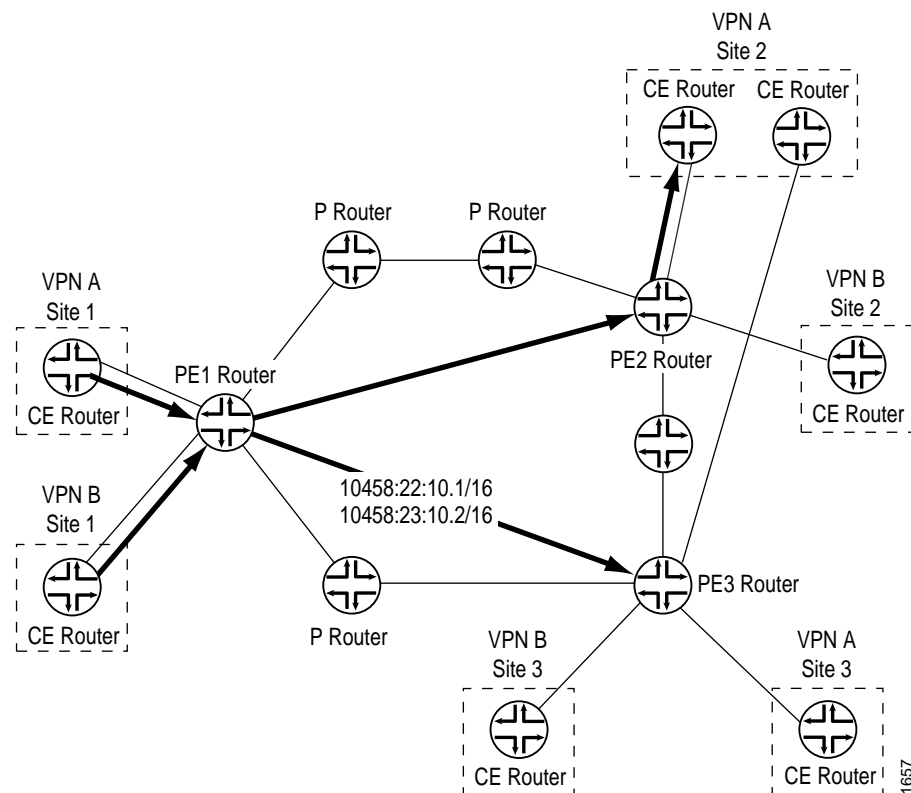
Route distinguishers are used only among PE routers to disambiguate IPv4 addresses from different VPNs. The ingress PE router creates a route distinguisher and converts IPv4 routes received from CE routers into VPN-IPv4 addresses. The egress PE routers convert VPN-IPv4 routes into IPv4 routes before announcing them to the CE router.

Because VPN-IPv4 addresses are a type of BGP address, you must configure IBGP sessions between pairs of PE routers so that the PE routers can distribute VPN-IPv4 routes within the provider's core network. (All PE routers are assumed to be within the same AS.)

You define BGP communities to constrain the distribution of routes among the PE routers. Defining BGP communities does not, by itself, disambiguate IPv4 addresses.

Figure 30 illustrates how Router PE1 adds the route distinguisher 10458:22:10.1/16 to routes received from the CE router at Site 1 in VPN A and forwards these routes to the other two PE routers. Similarly, Router PE1 adds the route distinguisher 10458:23:10.2/16 to routes received by the CE router at Site 1 in VPN B and forwards these routes to the other PE routers.

Figure 30: Route Distinguishers

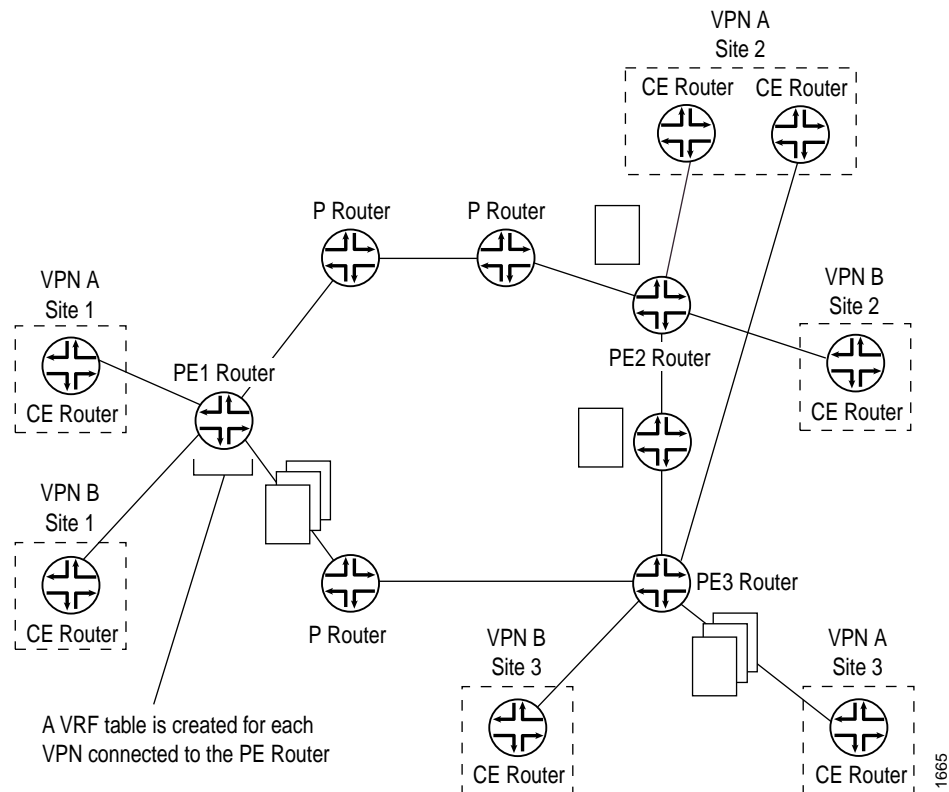


VPN Routing and Forwarding Tables

To separate a VPN's routes from routes in the public Internet or those in other VPNs, the PE router creates a separate routing table for each VPN, called a VPN Routing and Forwarding (VRF) table. The PE router creates one VRF table for each VPN that has a connection to a CE router. Any customer or site that belongs to the VPN can access only the routes in the VRF tables for that VPN.

Figure 31 illustrates the VRF tables that are created on the PE routers. The three PE routers have connections to CE routers that are in two different VPNs, so each of these PE routers creates two VRF tables, one for each VPN.

Figure 31: VRF Tables



Each VRF table is populated from routes received from directly connected CE sites associated with that VRF and from routes received from other PE routers that passed BGP community filtering and are in the same VPN.

Each PE router also maintains one global routing table (inet.0) to reach other routers in and outside the provider's core network.

Each customer connection (that is, logical interface) is associated with one VRF table. Only the VRF table associated with a customer site is consulted for packets from that site.

You can configure the router so that if a next hop to a destination is not found in the VRF table, the router performs a lookup in the global routing table, which is used for Internet access.

The JUNOS software uses the following routing tables for VPNs:

- **bgp.l3vpn.0**—Stores all VPN-IPv4 unicast routes received from other PE routers. (This table does not store routes received from directly connected CE routers.) This table is present only on PE routers.

When a PE router receives a route from another PE router, it places the route into its **bgp.l3vpn.0** routing table. The route is resolved using the information in the **inet.3** routing table. The resultant route is converted into IPv4 format and redistributed to all *routing-instance-name.inet.0* routing tables on the PE router if it matches the VRF import policy.

The `bgp.l3vpn.0` table is also used to resolve routes over the MPLS tunnels that connect the PE routers. These routes are stored in the `inet.3` routing table. PE-PE router connectivity must exist in `inet.3` (not just in `inet.0`) for VPN routes to be resolved properly.

To determine whether to add a route to the `bgp.l3vpn.0` routing table, the JUNOS software checks it against the VRF import policies for all the VPNs configured on the PE router. If the VPN-IPv4 route matches one of the policies, it is added to the `bgp.l3vpn.0` table. To display the routes in the `bgp.l3vpn.0` routing table, use the `show route table bgp.l3vpn.0` command.

- *routing-instance-name.inet.0*—Stores all unicast IPv4 routes received from directly connected CE routers in a routing instance (that is, in a single VPN) and all explicitly configured static routes in the routing instance. This is the VRF table and is present only on PE routers. For example, for a routing instance named VPN-A, the routing table for that instance is named `VPN-A.inet.0`.

When a CE router advertises to a PE router, the PE router places the route into the corresponding *routing-instance-name.inet.0* routing table and advertises the route to other PE routers if it passes a VRF export policy. Among other things, this policy tags the route with the route distinguisher (route target) that corresponds to the VPN site to which the CE belongs. A label is also allocated and distributed with the route. The `bgp.l3vpn.0` routing table is not involved in this process.

The *routing-instance-name.inet.0* table also stores routes announced by a remote PE router that match the VRF import policy for that VPN. The remote PE router redistributed these routes from its `bgp.l3vpn.0` table.

Routes are not redistributed from the *routing-instance-name.inet.0* table to the `bgp.l3vpn.0` table; they are directly advertised to other PE routers.

For each *routing-instance-name.inet.0* routing table, one forwarding table is maintained in the router's Packet Forwarding Engine. This table is maintained in addition to the forwarding tables that correspond to the router's `inet.0` and `mpls.0` routing tables. As with the `inet.0` and `mpls.0` routing tables, the best routes from the *routing-instance-name.inet.0* routing table are placed into the forwarding table.

To display the routes in the *routing-instance-name.inet.0* table, use the `show route table routing-instance-name.inet.0` command.

- `inet.3`—Stores all MPLS routes learned from LDP and RSVP signaling done for VPN traffic. The routing table stores the MPLS routes only if the `traffic-engineering bgp-igp` option is not enabled.

For VPN routes to be resolved properly, the `inet.3` table must contain routes to all the PE routers in the VPN.

To display the routes in the `inet.3` table, use the `show route inet.3` command.

- `inet.0`—Stores routes learned by the IBGP sessions between the PE routers. To provide Internet access to the VPN sites, configure the *routing-instance-name.inet.0* routing table to contain a default route to the `inet.0` routing table.

To display the routes in the `inet.0` table, use the `show route inet.0` command.

The following routing policies, which are defined in VRF import and export statements, are specific to VRF tables.

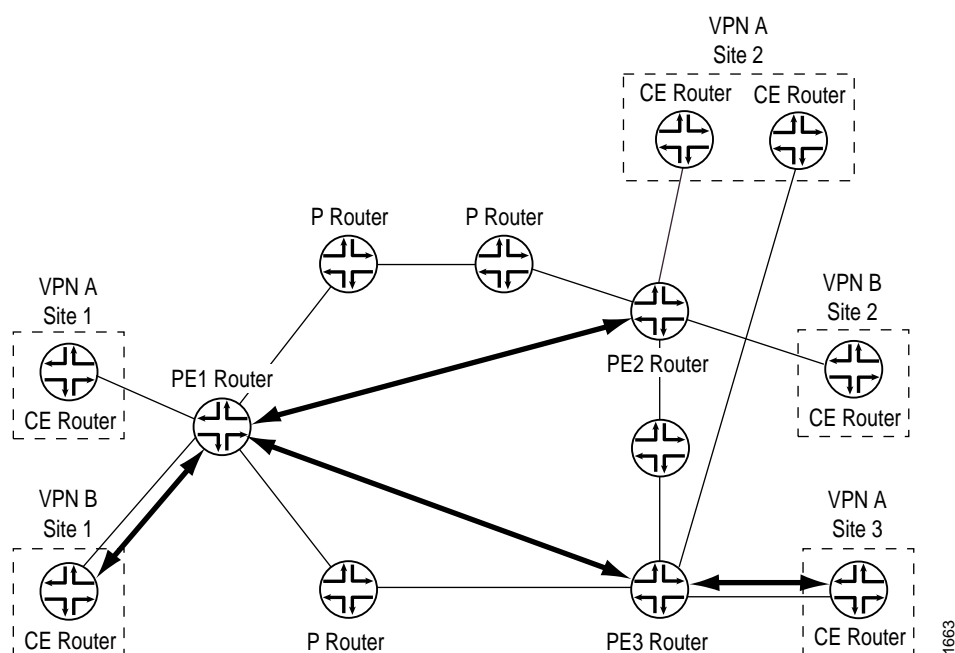
- Import policy—Applied to VPN-IPv4 routes learned from another PE router to determine whether the route should be added to the PE router's `bgp.l3vpn.0` routing table. Each routing instance on a PE router has a VRF import policy.
- Export policy—Applied to VPN-IPv4 routes that are announced to other PE routers. The VPN-IPv4 routes are IPv4 routes that have been announced by locally connected CE routers.

VPN route processing differs from normal BGP route processing in one way. In BGP, routes are accepted if they are not explicitly rejected by import policy. However, because many more VPN routes are expected, the JUNOS software does not accept (and hence store) VPN routes unless the route matches at least one VRF import policy. If no VRF import explicitly accepts the route, it is discarded and not even stored in the `bgp.l3vpn.0` table. As a result, if a VPN change occurs on a PE router—such as adding a new VRF table or changing a VRF import policy—the PE router sends a BGP route refresh message to the other PE routers (or to the route reflector if this is part of the VPN topology) to retrieve all VPN routes so they can be re-evaluated to determine whether they should be kept or discarded.

Route Distribution within a VPN

Within a VPN, the distribution of VPN-IPv4 routes occurs between the PE and CE routers and between the PE routers (see Figure 32).

Figure 32: Route Distribution within a VPN



This section discusses the following:

- Distribution of Routes from CE to PE Routers on page 195
- Distribution of Routes between PE Routers on page 196
- Distribution of Routes from PE to CE Routers on page 197

Distribution of Routes from CE to PE Routers

A CE router announces its routes to the directly connected PE router. The announced routes are in IPv4 format. The PE router places the routes into the VRF table for the VPN. In the JUNOS software, this is the *routing-instance-name.inet.0* routing table, where *routing-instance-name* is the configured name of the VPN.

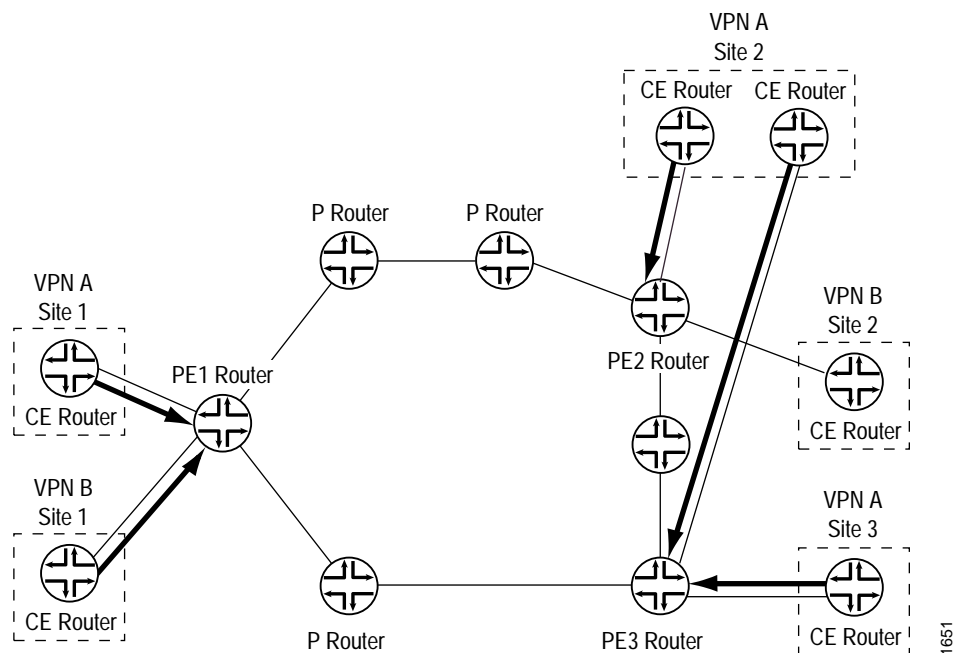
The connection between the CE and PE routers can be a remote connection (a WAN connection) or a direct connection (such as a Frame Relay or Ethernet connection).

CE routers can communicate with PE routers using one of the following routing protocols:

- OSPF
- RIP
- BGP
- Static route

Figure 33 illustrates how routes are distributed from CE routers to PE routers. Router PE1 is connected to two CE routers that are in different VPNs. Therefore, it creates two VRF tables, one for each VPN. The CE routers announce IPv4 routes. The PE router installs these routes into two different VRF tables, one for each VPN. Similarly, Router PE2 creates two VRF tables each into which they install the routes from their two directly connected CE routers. Router PE3 creates one VRF table because it is directly connected to only one VPN.

Figure 33: Distribution of Routes from CE Routers to PE Routers



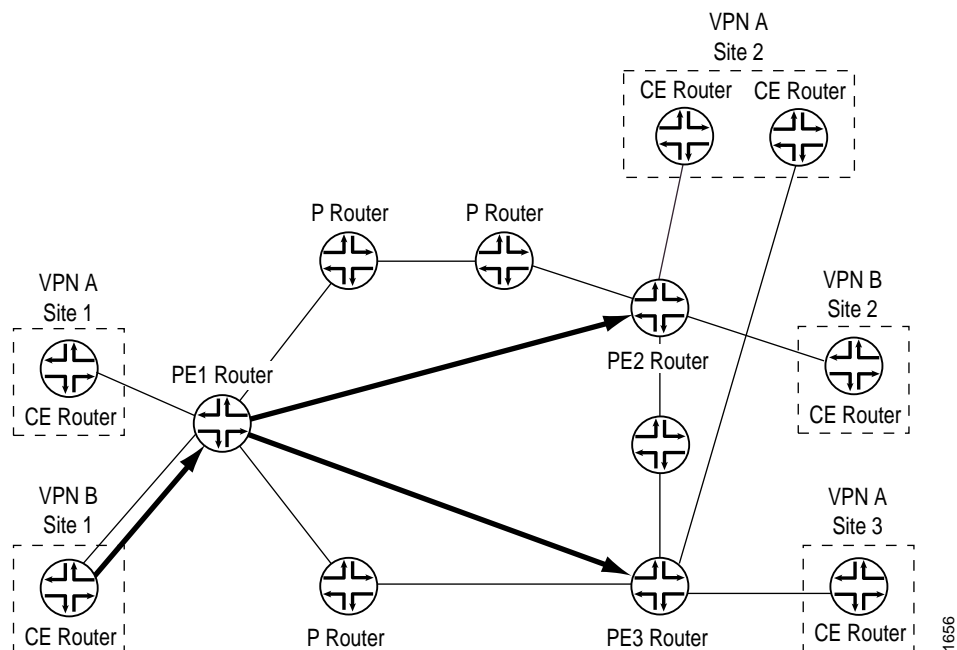
Distribution of Routes between PE Routers

When one PE router receives routes advertised from a directly connected CE router, it checks the received route against the VRF export policy for that VPN. If it matches, the route is converted to VPN-IPv4 format (that is, the route distinguisher [route target] is added to the route). The PE router then announces the route in VPN-IPv4 format to the remote PE routers. The routes are distributed using IBGP sessions, which are configured in the provider's core network. If the route does not match, it is not exported to other PE routers, but can still be used locally for routing, for example, if two CE routers in the same VPN are directly connected to the same PE router.

The remote PE router places the route into its `bgp.l3vpn.0` table if the route passes the import policy on the IBGP session between the PE routers. At the same time, it checks the route against the VRF import policy for the VPN. If it matches, the route distinguisher is removed from the route and it is placed into the VRF table (the `routing-instance-name.inet.0` table) in IPv4 format.

Figure 34 illustrates how Router PE1 distributes routes to the other PE routers in the provider's core network. Router PE2 and Router PE3 each have VRF import policies that they use to determine whether to accept routes received over the IBGP sessions and install them in their VRF tables.

Figure 34: Distribution of Routes between PE Routers



Distribution of Routes from PE to CE Routers

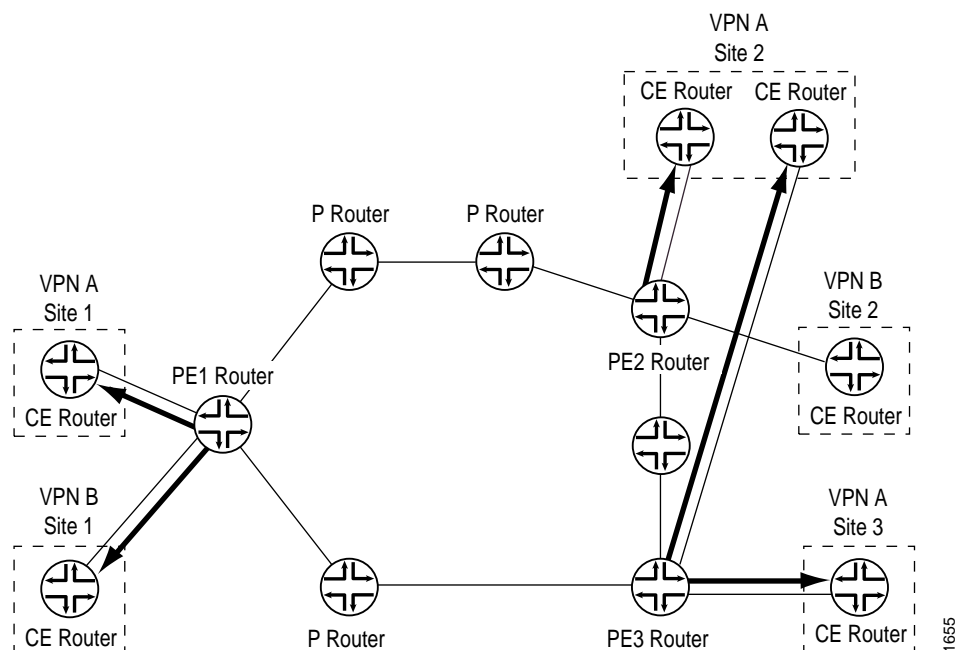
The remote PE router announces the routes in its VRF tables, which are in IPv4 format, to its directly connected CE routers.

PE routers can communicate with CE routers using one of the following routing protocols:

- OSPF
- RIP
- BGP
- Static route

Figure 35 illustrates how the three PE routers announce their routes to their connected CE routers.

Figure 35: Distribution of Routes from PE Routers to CE Routers



Forwarding across the Provider's Core Network

The PE routers in the provider's core network are the only routers that are configured to support VPNs and hence are the only routers that know about the existence of the VPNs. From the point of view of VPN functionality, the provider routers in the core—those provider routers that are not directly connected to CE routers—are merely routers along the tunnel between the ingress and egress PE routers.

The tunnels can be either LDP or MPLS. Any provider routers along the tunnel must support the protocol used for the tunnel, either LDP or MPLS.

When PE router-to-PE router forwarding is tunneled over MPLS LSPs, the MPLS packets have a two-level label stack (see Figure 36):

- Outer label—Label assigned to the address of the BGP next hop by the IGP next hop
- Inner label—Label that the BGP next hop assigned for the packet's destination address

Figure 36: Using MPLS LSPs to Tunnel between PE Routers

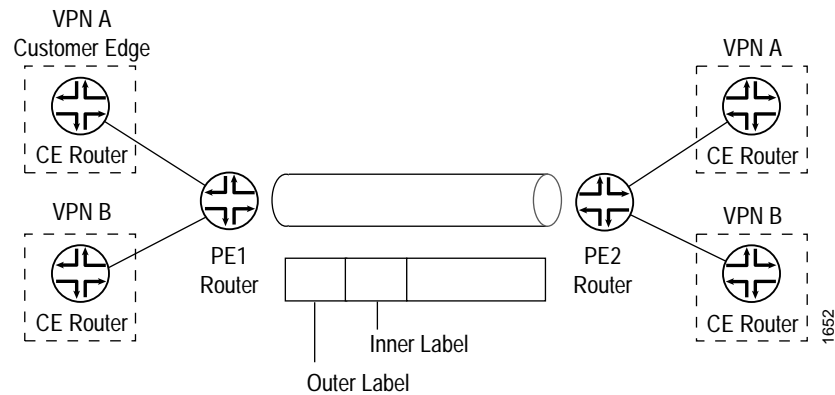
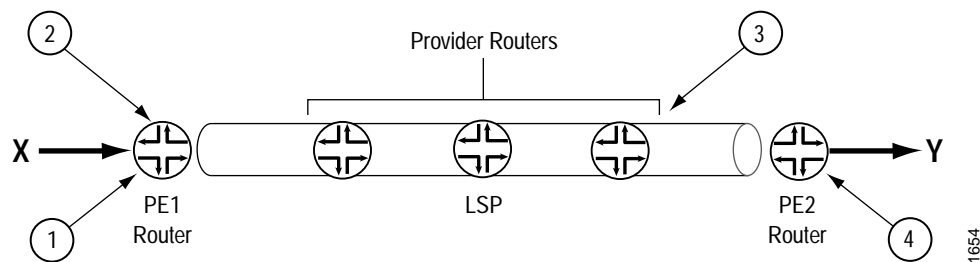


Figure 37 illustrates how the labels are assigned and removed:

1. When CE Router X forwards a packet to Router PE1 with a destination of CE Router Y, the PE route identifies the BGP next hop to Router Y and assigns a label that corresponds to the BGP next hop and identifies the destination CE router. This label is the inner label.
2. Router PE1 then identifies the IGP route to the BGP next hop and assigns a second label that corresponds to the LSP of the BGP next hop. This label is the outer label.
3. The inner label remains the same as the packet traverses the LSP tunnel. The outer label is swapped at each hop along the LSP and is then popped by the penultimate hop router (the third provider router).
4. Router PE2 pops the inner label from the route and forwards the packet to Router Y.

Figure 37: Label Stack



Routing Instances for VPNs

To implement Layer 3 VPNs in the JUNOS software, you configure one routing instance for each VPN. You configure the routing instances on PE routers only. Each VPN routing instance consists of the following components:

- VRF table—On each PE router, you configure one VRF table for each VPN.
- Set of interfaces that use the VRF table—The logical interface to each directly connected CE router must be associated with a VRF table. You can associate more than one interface with the same VRF table if more than one CE router in a VPN is directly connected to the PE router.
- Policy rules that control the import of routes into and the export of routes from the VRF table.
- One or more routing protocols that install routes from CE routers into the VRF table—You can use the BGP, OSPF, and RIP routing protocols, and you can use static routes.

Chapter 20

Layer 3 VPN Configuration Guidelines

To configure Layer 3 Virtual Private Network (VPN) functionality, you must enable VPN support on the provider edge (PE) router. You must also configure any provider (P) routers that service the VPN, and you must configure the customer edge (CE) routers so that their routes are distributed into the VPN.

To configure Layer 3 VPNs, you include statements at the [edit routing-instances] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    interface interface-name;
    instance-type vrf;
    route-distinguisher ( as-number:number | ip-address:number );
    vrf-import [ policy-name ];
    vrf-export [ policy-name ];
    protocols {
      bgp {
        bgp-configuration;
      }
      ospf {
        ospf-configuration;
      }
      rip {
        rip-configuration;
      }
    }
  }
  routing-options {
    autonomous-system autonomous-system <loops number>;
    forwarding-table {
      export [ policy-name ];
    }
    interface-routes {
      rib-group group-name;
    }
    martians {
      destination-prefix match-type <allow>;
    }
    options {
      syslog (level level | upto level);
    }
    rib routing-table {
      static {
        defaults {
          static-options;
        }
      }
    }
  }
}
```

```

    route destination-prefix {
        next-hop;
        static-options;
    }
}
martians {
    destination-prefix match-type <allow>;
}
static {
    defaults {
        static-options;
    }
    route destination-prefix {
        policy [ policy-name ];
        static-options;
    }
}
}
router-id address;
static {
    defaults {
        static-options;
    }
    route destination-prefix {
        policy [ policy-name ];
        static-options;
    }
}
}
}
}

```

For Layer 3 VPNs, only some of the statements in the [edit routing-instances] hierarchy are valid. For the full hierarchy, see *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

In addition to these statements, you must enable a signaling protocol, configure IBGP sessions between the PE routers, and configure an IGP on the PE and provider routers.

By default, Layer 3 VPNs are disabled.

This chapter describes the following tasks for configuring VPNs:

- Enable a Signaling Protocol on page 203
- Configure an IGP on PE and Provider Routers on page 206
- Configure an IBGP Session between PE Routers on page 206
- Configure Routing Instances for VPNs on PE Routers on page 207
- Configure VPN Routing between the PE and CE Routers on page 213

For configuration examples, see “Layer 3 VPN Configuration Examples” on page 227.

Enable a Signaling Protocol

For Layer 3 VPNs to function, you must enable a signaling protocol on the PE routers. You can do one of the following:

- Use LDP for VPN Signaling on page 203
- Use RSVP for VPN Signaling on page 204

Use LDP for VPN Signaling

To use LDP for VPN signaling, perform the following steps on the PE and provider routers:

1. Configure LDP on the interfaces in the core of the service provider's network by including the `ldp` statement at the [edit protocols] hierarchy level. You need to configure LDP only on the interfaces between PE routers or between PE and provider routers. You can think of these as the “core-facing” interfaces. You do not need to configure LDP on the interface between the PE and CE routers.

```
[edit]
protocols {
  ldp {
    interface interface-name;
  }
}
```

2. Configure the MPLS address family on the interfaces on which you enable LDP (that is, on the interfaces you configured in Step 1):

```
[edit]
interfaces {
  interface-name {
    unit logical-unit-number {
      family mpls;
    }
  }
}
```

Specify the interface name in the format *type-fpc/pic/port*.

3. Configure OSPF or IS-IS on each PE and provider router. You configure these protocols at the master instance of the routing protocol, not within the routing instance used for the VPN.

To configure OSPF, include the `ospf` statement at the [edit protocols] hierarchy level. At a minimum, you must configure a backbone area on at least one of the router's interfaces.

```
[edit]
protocols {
  ospf {
    area 0.0.0.0 {
      interface interface-name;
    }
  }
}
```

To configure IS-IS, include the `isis` statement at the `[edit protocols]` hierarchy level and configure the loopback interface and ISO family at the `[edit interfaces]` hierarchy level. At a minimum, you must enable IS-IS on the router, configure a network entity title (NET) on one of the router's interfaces (preferably the loopback interface, `lo0`), and configure the ISO family on all interfaces on which you want IS-IS to run. When you enable IS-IS, Level 1 and Level 2 are enabled by default. The following is the minimum IS-IS configuration. In the address statement, *address* is the NET.

```
[edit]
interfaces {
  lo0 {
    unit logical-unit-number {
      family iso {
        address address;
      }
    }
  }
  type-fpc/pic/port {
    unit logical-unit-number {
      family iso;
    }
  }
}
protocols {
  isis {
    interface all;
  }
}
```

For more information about configuring OSPF and IS-IS, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Use RSVP for VPN Signaling

To use RSVP for VPN signaling, perform the following steps:

1. On each PE router, configure traffic engineering. To do this, you must configure an IGP that supports traffic engineering (either IS-IS or OSPF) and enable traffic engineering support for that protocol.

To enable OSPF traffic engineering support, include the `traffic-engineering` statement at the `[edit protocols ospf]` hierarchy level:

```
[edit protocols ospf]
traffic-engineering {
  no-topology;
  shortcuts;
}
```

For IS-IS, traffic engineering support is enabled by default.

2. On each PE and provider router, enable RSVP on the router interfaces that participate in the label-switched path (LSP). On the PE router, these are the interfaces that are the ingress and egress points to the LSP. On the provider router, these are the interfaces that connect the LSP between the PE routers. Do not enable RSVP on the interface between the PE and the CE routers, because this interface is not part of the LSP.

To configure RSVP on the PE and provider routers, include the interface statement at the [edit rsvp] hierarchy level. Include one interface statement for each interface on which you are enabling RSVP.

```
[edit]
rsvp {
  interface interface-name;
}
```

3. On each PE router, configure an MPLS LSP to the PE router that is the LSP's egress point. To do this, include the label-switched-path and interface statements at the [edit mpls] hierarchy level.

```
[edit]
mpls {
  label-switched-path path-name {
    to ip-address;
  }
  interface interface-name;
}
```

In the to statement, specify the address of the LSP's egress point, which is an address on the remote PE router.

In the interface statement, specify the name of the interface (both the physical and logical portions). Include one interface statement for the interface associated with the LSP.

When you configure the same interface at the [edit interfaces] hierarchy level, you must also configure family mpls and family inet when configuring the logical interface:

```
[edit interfaces]
interface-name {
  unit logical-unit-number {
    family inet;
    family mpls;
  }
}
```

4. On all provider routers that participate in the LSP, enable MPLS by including the interface statement at the [edit mpls] hierarchy level. Include one interface statement for each connection to the LSP.

```
[edit]
mpls {
  interface interface-name;
  interface interface-name;
}
```

5. Enable MPLS on the interface between the PE and CE routers by including the interface statement at the [edit mpls] hierarchy level. Doing this allows the PE router to assign an MPLS label to traffic entering the LSP or to remove the label from traffic exiting the LSP.

```
[edit]
mpls {
  interface interface-name;
}
```

For information about configuring MPLS, see “Configure MPLS Signaled LSPs” on page 43, “Configure Static LSPs” on page 73, and “Configure Explicit-Path LSPs” on page 79. For information about configuring RSVP, see “RSVP Configuration Guidelines” on page 121.

Configure an IGP on PE and Provider Routers

To allow the PE and provider routers to exchange routing information, you must either configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the routing instance used for the VPN (that is, not at the [edit routing-instances] hierarchy level).

When you configure the PE router, do not configure any summarization of the PE router's loopback addresses at the area boundary. Each PE router's loopback address should appear as a separate route.

For information about configuring routing protocols and static routes, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Configure an IBGP Session between PE Routers

You must configure an IBGP session between PE routers to allow the PE routers to exchange information about routes originating and terminating in the VPN. To do this, include the family inet-vpn statement when configuring IBGP:

```
[edit protocols]
bgp {
  group group-name {
    type internal;
    local-address ip-address;
    family inet-vpn {
      unicast;
    }
    neighbor ip-address;
  }
}
```

The family inet-vpn statement indicates that the IBGP session is for the VPN.

The IP address in the local-address statement is the address of the loopback interface (lo0) on the local PE router. The IBGP session for VPNs runs through the loopback address. (You must also configure the lo0 interface at the [edit interfaces] hierarchy level.)

The IP address in the neighbor statement is the loopback address of the neighboring PE router. If you are using RSVP signaling, this IP address is the same address you specify in the to statement at the [edit mpls label-switched-path] hierarchy level when you configure the MPLS LSP.

Configure Routing Instances for VPNs on PE Routers

To configure routing instances for VPNs, include the `routing-instances` statement at the [edit] hierarchy level. You configure VPN routing instances only on PE routers.

```
[edit]
routing-instances {
  routing-instance-name {
    instance-type vrf;
    interface interface-name;
    route-distinguisher ( as-number:number | ip-address:number );
    vrf-import [ policy-name ];
    vrf-export [ policy-name ];
  }
}
```



Note

For the VPN to function, you must include the instance-type, interface, route-distinguisher, vrf-import, and vrf-export statements in the routing instance configuration on the PE router.

The following sections describe how to configure VPN routing instances:

- Configure the Instance Type on page 207
- Configure Interfaces for VPN Routing on page 207
- Configure the Route Distinguisher on page 208
- Configure Policy for the PE Router's VRF Table on page 209

Configure the Instance Type

Each PE router uses a VPN Routing and Forwarding (VRF) table for distributing routes within the VPN. To create the VRF table on the PE router, include the `instance-type` statement at the [edit routing-instances *routing-instance-name*] hierarchy level, specifying the instance type as `vrf`:

```
[edit routing-instances routing-instance-name]
instance-type vrf;
```

Configure Interfaces for VPN Routing

On each PE router, you must configure an interface over which the VPN traffic travels between the PE and CE routers. To do this, include the `interface` statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
interface interface-name;
```

You should specify both the physical and logical portions of the interface name, in the following format:

physical.logical

For example, in so-1/2/1.0, so-1/2/1 is the physical portion of the interface name and 0 is the logical portion. If you do not specify the logical portion of the interface name, 0 is used.

A logical interface can be associated with only one routing instance.

When you configure the same interface at the [edit interfaces] hierarchy level, you must also configure family inet and family mpls when configuring the logical interface:

```
[edit interfaces]
interface-name {
  unit logical-unit-number {
    family inet;
    family mpls;
  }
}
```



Note

If you enable a routing protocol on all instances by specifying interfaces all when configuring the master instance of the protocol at the [edit protocols] hierarchy level and if you configure a specific interface for VPN routing at the [edit routing-instances *routing-instance-name*] hierarchy level, the latter interface statement takes precedence and the interface is used exclusively for the VPN.

If you explicitly configure the same interface name at both the [edit protocols] and [edit routing-instances *routing-instance-name*] hierarchy levels, when you try to commit the configuration, it will fail.

Configure the Route Distinguisher

Each routing instance that you configure on a PE router must have a unique route distinguisher associated with it. The route distinguisher is used to place bounds around a VPN so the same IP address prefixes can be used in different VPNs without overlapping.

We recommend that you use unique route distinguishers for each routing instance that you configure. Although you could use the same route distinguisher on all PE routers in the same VPN, if you use a unique route distinguisher, you can determine the PE router from which a route originated.

To configure a route distinguisher on a PE router, include the route-distinguisher statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
route-distinguisher ( as-number:number | ip-address:number );
```

The route distinguisher is a 6-byte value that you can specify in one of the following formats:

- *as-number:number*, where *as-number* is an AS number (a 2-byte value) and *number* is any 4-byte value. The AS number can be in the range 1 through 65535. We recommend that you use an IANA assigned, nonprivate AS number, preferably the ISP's own or the customer's own AS number.
- *ip-address:number*, where *ip-address* is an IP address (a 4-byte value) and *number* is any 2-byte value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range.

Configure Policy for the PE Router's VRF Table

On each PE router, you must define policies that define how routes are imported into and exported from the router's VRF table. In these policies, you must define the route target and you can optionally define the route origin.

The following sections describe how to configure policy for the VRF tables:

- Configure the Route Target on page 209
- Configure the Route Origin on page 210
- Configure Import Policy for the PE Router's VRF Table on page 210
- Configure Export Policy for the PE Router's VRF Table on page 211

Configure the Route Target

In the import and export policies for the PE router's VRF table, you must define the route target, which defines which VPN the route is part of. To do this, include the target option in the community statement at the [edit policy-options] hierarchy level:

```
[edit policy-options]
community name members target: community-id;
```

name is the name of the community.

community-id is the identifier of the community. You specify it in one of the following formats:

- *as-number:number*, where *as-number* an AS number (a 2-byte value) and *number* is a 4-byte community identifier. The AS number can be in the range 1 through 65535. We recommend that you use an IANA assigned, nonprivate AS number, preferably the ISP's own or the customer's own AS number. The community identifier can be a number in the range 0 through $2^{32} - 1$.
- *ip-address:number*, where *ip-address* is an IPv4 address (a 4-byte value) and *number* is a 2-byte community identifier. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range. The community identifier can be a number in the range 1 through 65535.

Configure the Route Origin

In the import and export policies for the PE router's VRF table, you can optionally define the route origin (otherwise known as the site of origin), which identifies the set of routes learned from a particular CE site. To do this, include the origin option in the community statement at the [edit policy-options] hierarchy level:

```
[edit policy-options]
community name members origin: community-id;
```

name is the name of the community.

community-id is the identifier of the community. You specify it in one of the following format:

- *as-number:number*, where *as-number* is an AS number (a 2-byte value) and *number* is a 4-byte community identifier. The AS number can be in the range 1 through 65535. We recommend that you use an IANA assigned, nonprivate AS number, preferably the ISP's own or the customer's own AS number. The community identifier can be a number in the range 0 through $2^{32} - 1$.
- *ip-address:number*, where *ip-address* is an IPv4 address (a 4-byte value) and *number* is a 2-byte community identifier. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range. The community identifier can be a number in the range 1 through 65535.

Configure Import Policy for the PE Router's VRF Table

Each VPN must have a policy that defines how routes are imported into the PE router's VRF table. The import policy is applied to routes received from other PE routers in the VPN. The policy must evaluate all routes received over the IBGP session with the other PE router. If the routes match the conditions, the route is installed in the PE router's *routing-instance-name*.inet.0 VRF table. The import policy must contain a second term that rejects all other routes.

Unless the import policy contains only a then reject statement, it must include a reference to a community. Otherwise, when you try to commit the configuration, the commit fails.

To configure an import policy for the PE router's VRF table, follow these steps:

1. To define the import policy, include the `policy-statement` statement at the `[edit policy-options]` hierarchy level. For all PE routers, the import policy must always include the following, at a minimum:

```
[edit]
policy-options {
  policy-statement import-policy-name {
    term import-term-name {
      from {
        protocol bgp;
        community community-id;
      }
      then accept;
    }
    term term-name {
      then reject;
    }
  }
}
```

The *import-policy-name* policy evaluates all routes received over the IBGP session with the other PE router. If the routes match the conditions in the from statement, the route is installed in the PE router's *routing-instance-name*.inet.0 VRF table. The second term in the policy rejects all other routes.

For more information about creating policies, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

2. To apply the import policy, include the `vrf-import` statement at the `[edit routing-instances routing-instance-name]` hierarchy level:

```
[edit routing-instances routing-instance-name]
vrf-import [ import-policy-name ];
```

Note that you can configure multiple import policies.

Configure Export Policy for the PE Router's VRF Table

Each VPN must have a policy that defines how routes are exported from the PE router's VRF table. The export policy is applied to routes sent to other PE routers in the VPN. The export policy must evaluate all routes received over the routing protocol session with the CE router. (This session can use either the BGP, OSPF, or RIP routing protocol or static routes.) If the routes match the conditions, the specified community target (which is the route target) is added to them and they are exported to the remote PE routers. The export policy must contain a second term that rejects all other routes.

Export policies defined within the VPN routing instance are the only export policies that apply to the VRF table. Any export policy that you define on the IBGP session between the PE routers has no effect on the VRF table.

To configure an export policy for the PE router's VRF table, follow these steps:

1. To define the export policy, include the `policy-statement` statement at the `[edit policy-options]` hierarchy level. For all PE routers, the export policy must distribute VPN routes to and from the connected CE routers in accordance with the type of routing protocol that you configure between the CE and PE routers within the routing instance. The export policy must always include the following, at a minimum:

```
[edit]
policy-options {
  policy-statement export-policy-name {
    term export-term-name {
      from protocol (bgp | ospf | rip | static);
      then {
        community add community-id;
        accept;
      }
    }
    term term-name {
      then reject;
    }
  }
}
```

The *export-policy-name* policy evaluates all routes received over the routing protocol session with the CE router. (This session can use either the BGP, OSPF, or RIP routing protocol or static routes.) If the routes match the conditions in the `from` statement, the community target specified in the `then community add` statement is added to them and they are exported to the remote PE routers. The second term in the policy rejects all other routes.

For more information about configuring routing within the routing instance, see [Configure VPN Routing between the PE and CE Routers](#) on page 213. For more information about creating policies, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

2. To apply the policy, include the `vrf-export` statement at the `[edit routing-instances routing-instance-name]` hierarchy level:

```
[edit routing-instances routing-instance-name]
vrf-export [ export-policy-name ];
```

Note that you can configure multiple export policies.

Configure VPN Routing between the PE and CE Routers

For the PE router to distribute VPN-related routes to and from connected CE routers, you must configure routing within the VPN routing instance. You can configure a routing protocol—either BGP, OSPF, or RIP—or you can configure static routing. For the connection to each CE router, you can configure only one type of routing.

This section describes how to configure the following protocols:

- Configure BGP between the PE and CE Routers on page 213
- Configure OSPF between the PE and CE routers on page 213
- Configure RIP between the PE and CE Routers on page 214
- Configure Static Routes between the PE and CE Routers on page 214

Configure BGP between the PE and CE Routers

To configure BGP as the routing protocol between the PE and the CE router, include the protocols bgp statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    protocols {
      bgp {
        group group-name {
          peer-as as-number;
          neighbor ip-address;
        }
      }
    }
  }
}
```

Configure OSPF between the PE and CE routers

You can configure OSPF to distribute VPN-related routes between the PE and CE routers. As part of the OSPF configuration for VPNs, you need to configure an OSPF domain ID for each distinct OSPF domain. Routes from an OSPF domain need to have an OSPF domain ID when they are distributed in BGP as VPN-IPv4 routes in VPNs with multiple OSPF domains. In a VPN connecting multiple OSPF domains, there is a possibility that the routes from one of the domains could overlap with the routes of a different domain. Configuring a unique OSPF domain ID for each domain ensures that the routes for each domain remain separate.

When a PE router receives a route with a different OSPF domain ID, it redistributes the route as a type 5 LSA. If the OSPF domain IDs match and the route is a summary route, it is distributed as a type 3 LSA (type 5 LSAs are passed as type 5 LSAs).

Each VRF table in a PE router associated with an OSPF instance is configured with the same OSPF domain ID.

By default, an OSPF domain ID is not configured. The PE router distributes summary routes as type 3 LSAs and assumes that only one OSPF domain exists.

To configure OSPF as the routing protocol between the PE and the CE router, include the protocols ospf statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    protocols {
      ospf {
        area area {
          interface interface-name;
        }
        domain-id domain ID;
      }
    }
  }
}
```

Configure RIP between the PE and CE Routers

To configure RIP as the routing protocol between the PE and the CE router, include the protocols rip statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    protocols {
      rip {
        group group-name {
          neighbor interface-name;
        }
      }
    }
  }
}
```

Configure Static Routes between the PE and CE Routers

To configure a static route between the PE and the CE router, include the routing-options static statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    routing-options {
      static {
        route destination-prefix {
          next-hop;
          static-options;
        }
      }
    }
  }
}
```

For more information about configuring routing protocols and static routes, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Chapter 21

Layer 3 VPN Configuration Troubleshooting Guidelines

This chapter discusses the following strategies and tools for troubleshooting Layer 3 Virtual Private Network (VPN) configurations:

- Diagnose Common Problems on page 215
- Use the Ping and Traceroute Commands to Troubleshoot Layer 3 VPN Topologies on page 219

Diagnose Common Problems

When problems arise in a Layer 3 VPN configuration, the best way to troubleshoot is to start at one end of the VPN (that is, at the local CE router) and follow the routes to the other end of the VPN (that is, the remote CE router). The following troubleshooting steps should help you diagnose common problems:

1. If you have configured a routing protocol between the local PE and CE routers, check that the peering and adjacency is fully operational. When you do this, make sure to specify the name of the routing instance. For example, if you are checking OSPF adjacencies, you would enter the command `show ospf neighbor instance routing-instance-name` on the PE router.

If the peering and adjacency are not fully operational, check the routing protocol configuration on the CE router and check the routing protocol configuration for the associated VPN routing instance on the PE router.

2. Check that the local CE and PE routers can ping each other.

To check that the local CE router can ping the VPN interface on the local PE router, use a ping command in the following format, specifying the IP address or name of the PE router:

```
ping (ip-address | host-name)
```

To check that the local PE router can ping the CE router, use a ping command in the following format, specifying the IP address or name of the CE router, the name of the interface used for the VPN, and the source IP address (the local address) in outgoing ECHO_REQUEST packets:

```
ping ip-address vpn-interface interface local echo-address
```

Often, the peering or adjacency between the local CE and local PE routers needs to come up before a ping command is successful. To check that a link is operational in a lab setting, remove the interface from the VRF by deleting the interface statement from the [edit routing-instance *routing-instance-name*] hierarchy level and recommitting the configuration. Doing this removes the interface from the VPN. Then try the ping command again. If the command is successful, configure the interface back into the VPN and check the routing protocol configuration on the local CE and PE routers again.

3. On the local PE router, check that the routes from the local CE router are in the VRF routing table (*routing-instance-name*.inet.0):

```
show route table routing-instance-name.inet.0 [detail]
```

The following example shows what the routing table entries look like. Here, the loopback address of the CE router is 10.255.14.155/32 and the routing protocol between the PE and CE routers is BGP. The entry looks like any ordinary BGP announcement.

```
10.255.14.155/32 (1 entry, 1 announced)
  *BGP      Preference: 170/-101
            Nexthop: 192.168.197.141 via fe-1/0/0.0, selected
            State: <Active Ext>
            Peer AS:      1
            Age: 45:46
            Task: BGP_1.192.168.197.141+179
            Announcement bits (2): 0-BGP.0.0.0.0+179 1-KRT
            AS path: 1 I
            Localpref: 100
            Router ID: 10.255.14.155
```

If the routes from the local CE router are not present in the VRF routing table, check that the CE router is advertising routes to the PE router. If static routing is used between the CE and PE routers, make sure the proper static routes are configured.

4. On a remote PE router, check that the routes from the local CE router are present in the bgp.l3vpn.0 routing table:

```
show route table bgp.l3vpn.0 extensive
```

The following example shows what the routing table entries look like.

```
10.255.14.175:3:10.255.14.155/32 (1 entry, 0 announced)
  *BGP      Preference: 170/-101
            Route Distinguisher: 10.255.14.175:3
            Source: 10.255.14.175
            Nexthop: 192.168.192.1 via fe-1/1/2.0, selected
            label-switched-path vpn07-vpn05
            Push 100004, Push 100005(top)
            State: <Active Int Ext>
            Local AS:      69 Peer AS:      69
            Age: 15:27      Metric2: 338
            Task: BGP_69.10.255.14.175+179
            AS path: 1 I
            Communities: target:69:100
            BGP next hop: 10.255.14.175
            Localpref: 100
            Router ID: 10.255.14.175
            Secondary tables: VPN-A.inet.0
```

The output of the `show route table bgp.l3vpn.0` extensive command contains the following information specific to the VPN:

- In the prefix name (the first line of the output), the route distinguisher is added to the route prefix of the local CE router. Because the route distinguisher is unique within the Internet, the concatenation of the route distinguisher and IP prefix provides unique VPN-IPv4 routing entries.
- The Route Distinguisher field lists the route distinguisher separately from the VPN-IPv4 address.
- The label-switched-path field shows the name of the LSP used to carry the VPN traffic.
- The Push field shows both labels being carried in the VPN-IPv4 packet. The first label is the inner label, which is the VPN label that was assigned by the PE router. The second label is the outer label, which is an RSVP label.
- The Communities field lists the target community.
- The Secondary tables field lists other routing tables on this router into which this route has been installed.

If routes from the local CE router are not present in the `bgp.l3vpn.0` routing table on the remote PE router, do the following:

- Check the VRF import filter on the remote PE router, which is configured in the `vrf-import` statement. (On the local PE router, you check the VRF export filter, which is configured with the `vrf-export` statement.)
- Check that there is an operational LSP or an LDP path between the PE routers. To do this, check that the IBGP next-hop addresses are in the `inet.3` table.
- Check that the IBGP session between the PE routers is established and configured properly.
- Check for “hidden” routes, which usually means that routes were not labeled properly. To do this, use the `show route table bgp.l3vpn.0 hidden` command.
- Check that the inner label matches the inner VPN label that is assigned by the local PE router. To do this, use the `show route table mpls` command.

The following example shows the output of this command on the remote PE router. Here, the inner label is 100004.

```
...
Push 100004, Push 10005 (top)
```

The following example shows the output of this command on the local PE router, which shows that the inner label of 100004 matches the inner label on the remote PE router:

```
...
100004          *[VPN/7] 06:56:25, metric 1
                > to 192.168.197.141 via fe-1/0/0.0, Pop
```

5. On the remote PE router, check that the routes from the local CE router are present in the VRF table (*routing-instance-name.inet.0*):

```
show route table routing-instance-name.inet.0 [detail]
```

The following example shows what the routing table entries look like.

```
10.255.14.155/32 (1 entry, 1 announced)
  *BGP      Preference: 170/-101
            Route Distinguisher: 10.255.14.175:3
            Source: 10.255.14.175
            Nexthop: 192.168.192.1 via fe-1/1/2.0, selected
            label-switched-path vpn07-vpn05
            Push 100004, Push 100005(top)
            State: <Secondary Active Int Ext>
            Local AS: 69 Peer AS: 69
            Age: 1:16:22 Metric2: 338
            Task: BGP_69.10.255.14.175+179
            Announcement bits (2): 1-KRT 2-VPN-A-RIP
            AS path: 1 I
            Communities: target:69:100
            BGP next hop: 10.255.14.175
            Localpref: 100
            Router ID: 10.255.14.175
            Primary Routing Table bgp.l3vpn.0
```

In this routing table, the route distinguisher is no longer prepended to the prefix. The last line, Primary Routing Table, lists the table from which this route was learned.

If the routes are not present in this routing table, but were present in Step 4, the routes might have not passed the VRF import policy on the remote PE router.

If a VPN-IPv4 route matches no vrf-import policy, the route does not show up in the bgp.l3vpn table at all and hence is not present in the VRF table. If this occurs, it might indicate that on the PE router, you have configured another vrf-import statement on another VPN (with a common target), and the routes show up in the bgp.l3vpn.0 table, but are imported into the wrong VPN.

6. On the remote CE router, check that the routes from the local CE router are present in the routing table (inet.0):

```
show route
```

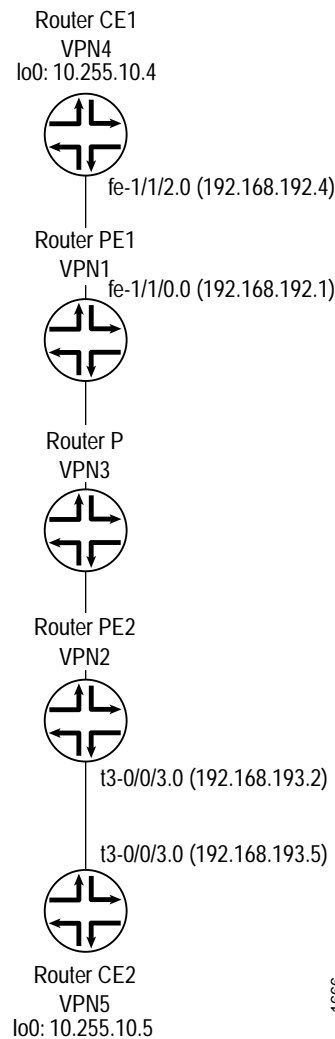
If the routes are not present, check the routing protocol configuration between the remote PE and CE routers, and make sure that peers and adjacencies (or static routes) between the PE and CE routers are correct.

7. If, in Steps 1 through 6, you have determined that routes originated from the local CE router are correct, check the routes originated from the remote CE router by repeating Steps 1 through 6.

Use the Ping and Traceroute Commands to Troubleshoot Layer 3 VPN Topologies

This section provides examples of how to use the ping command to check the accessibility of various routers in a VPN topology and how to use the traceroute command to check the path that packets travel between the VPN routers. The topology shown in Figure 38 is used to illustrate these commands.

Figure 38: Layer 3 VPN Topology for Ping and Traceroute Command Examples



1666

Ping One CE Router from the Other

You can ping one CE router from the other by specifying the other CE router's loopback address as the IP address in the ping command. This ping command succeeds if the loopback addresses have been announced by the CE routers to their directly connected PE routers. The success of these ping commands also means that Router CE1 can ping any network devices beyond Router CE2, and vice versa. See Figure 38 for the topology referenced in these examples.

Ping Router CE2 (VPN5) from Router CE1 (VPN4):

```
user@vpn4> ping 10.255.10.5 local 10.255.10.4 count 3
PING 10.255.10.5 (10.255.10.5): 56 data bytes
64 bytes from 10.255.10.5: icmp_seq=0 ttl=253 time=1.086 ms
64 bytes from 10.255.10.5: icmp_seq=1 ttl=253 time=0.998 ms
64 bytes from 10.255.10.5: icmp_seq=2 ttl=253 time=1.140 ms

--- 10.255.10.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.998/1.075/1.140/0.059 ms
```

To determine the path from Router CE1's loopback interface to Router CE2's loopback interface, use the following traceroute command:

```
user@vpn4> traceroute 10.255.10.5 source 10.255.10.4
traceroute to 10.255.10.5 (10.255.10.5) from 10.255.10.4, 30 hops max, 40 byte
packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.680 ms  0.491 ms  0.456 ms
 2  vpn2-t3-001.isp-core.net (192.168.192.110) 0.857 ms  0.766 ms  0.754 ms
    MPLS Label=100005 CoS=0 TTL=1 S=1
 3  vpn5.isp-core.net (10.255.10.5)  0.825 ms  0.886 ms  0.732 ms
```

Ping Router CE1 (VPN4) from Router CE2 (VPN5):

```
user@vpn5> ping 10.255.10.4 local 10.255.10.5 count 3
PING 10.255.10.4 (10.255.10.4): 56 data bytes
64 bytes from 10.255.10.4: icmp_seq=0 ttl=253 time=1.042 ms
64 bytes from 10.255.10.4: icmp_seq=1 ttl=253 time=0.998 ms
64 bytes from 10.255.10.4: icmp_seq=2 ttl=253 time=0.954 ms

--- 10.255.10.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.954/0.998/1.042/0.036 ms
```

To determine the path from Router CE2 to Router CE1, use the following traceroute command:

```
user@vpn5> traceroute 10.255.10.4 source 10.255.10.5
traceroute to 10.255.10.4 (10.255.10.4) from 10.255.10.5, 30 hops max, 40 byte
packets
 1  vpn-08-t3-003.isp-core.net (192.168.193.2)  0.686 ms  0.519 ms  0.548 ms
 2  vpn1-so-100.isp-core.net (192.168.192.100) 0.918 ms  0.869 ms  0.859 ms
    MPLS Label=100021 CoS=0 TTL=1 S=1
 3  vpn4.isp-core.net (10.255.10.4)  0.878 ms  0.760 ms  0.739 ms
```

Ping the Remote PE and CE Routers from the Local CE Router

From the local CE router, you can ping the VPN interfaces on the remote PE and CE routers, which are point-to-point interfaces. See Figure 38 for the topology referenced in these examples.

Ping Router CE2 (VPN5) from Router CE1 (VPN4):

```
user@vpn4> ping 192.168.193.5 local 10.255.10.4 count 3
PING 192.168.193.5 (192.168.193.5): 56 data bytes
64 bytes from 192.168.193.5: icmp_seq=0 ttl=253 time=1.040 ms
64 bytes from 192.168.193.5: icmp_seq=1 ttl=253 time=0.891 ms
64 bytes from 192.168.193.5: icmp_seq=2 ttl=253 time=0.944 ms

--- 192.168.193.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.891/0.958/1.040/0.062 ms
```

To determine the path from Router CE1's loopback interface to Router CE2's directly connected interface, use the following traceroute command:

```
serpil@vpn4> traceroute 192.168.193.5 source 10.255.10.4
traceroute to 192.168.193.5 (192.168.193.5) from 10.255.10.4, 30 hops max, 40
byte packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.669 ms  0.508 ms  0.457 ms
 2  vpn2-t3-001.isp-core.net (192.168.192.110) 0.851 ms  0.769 ms  0.750 ms
    MPLS Label=100000 CoS=0 TTL=1 S=1
 3  vpn5-t3-003.isp-core.net (192.168.193.5)  0.829 ms  0.838 ms  0.731 ms
```

Ping Router PE2 (VPN2) from Router CE1 (VPN4). In this case, packets that originate at Router CE1 go to Router PE2, then to Router CE2, and back to Router PE2 before Router PE2 can respond to ICMP requests. You can verify this using the traceroute command.

```
user@vpn4> ping 192.168.193.2 local 10.255.10.4 count 3
PING 192.168.193.2 (192.168.193.2): 56 data bytes
64 bytes from 192.168.193.2: icmp_seq=0 ttl=254 time=1.080 ms
64 bytes from 192.168.193.2: icmp_seq=1 ttl=254 time=0.967 ms
64 bytes from 192.168.193.2: icmp_seq=2 ttl=254 time=0.983 ms

--- 192.168.193.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.967/1.010/1.080/0.050 ms
```

To determine the path from Router CE1 to Router PE2, use the following traceroute command:

```
user@vpn4> traceroute 192.168.193.2 source 10.255.10.4
traceroute to 192.168.193.2 (192.168.193.2) from 10.255.10.4, 30 hops max, 40
byte packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.690 ms  0.490 ms  0.458 ms
 2  vpn2-t3-003.isp-core.net (192.168.193.2) 0.846 ms  0.768 ms  0.749 ms
    MPLS Label=100000 CoS=0 TTL=1 S=1
 3  vpn5-t3-003.isp-core.net (192.168.193.5)  0.643 ms  0.703 ms  0.600 ms
 4  vpn-08-t3-003.isp-core.net (192.168.193.2) 0.810 ms  0.739 ms  0.729 ms
```

You cannot ping one CE router from the other if the VPN interface is a multiaccess interface, such as the fe-1/1/2.0 interface on Router CE1. To ping Router CE1 from Router CE2, on Router PE1, you must configure a static route to the VPN interface of Router CE1 that has a next hop pointing to Router CE1 (at the [edit routing-instance *routing-instance-name*] hierarchy level) and this route must be announced from Router PE1 to Router PE2. The following configuration portions illustrate this configuration:

```
[edit]
routing-instances {
  direct-multipoint {
    instance-type vrf;
    interface fe-1/1/0.0;
    route-distinguisher 69:1;
    vrf-import direct-import;
    vrf-export direct-export;
    routing-options {
      static {
        route 192.168.192.4/32 next-hop 192.168.192.4;
      }
    }
  }
  protocols {
    bgp {
      group to-vpn4 {
        peer-as 1;
        neighbor 192.168.192.4;
      }
    }
  }
}
policy-options {
  policy-statement direct-export {
    term a {
      from protocol bgp;
      then {
        community add direct-comm;
        accept;
      }
    }
    term b {
      from {
        protocol static;
        route-filter 192.168.192.4/32 exact;
      }
      then {
        community add direct-comm;
        accept;
      }
    }
    term d {
      then reject;
    }
  }
}
```


Now you can ping Router CE1 from Router CE2:

```
user@vpn5> ping 192.168.192.4 local 10.255.10.5 count 3
PING 192.168.192.4 (192.168.192.4): 56 data bytes
64 bytes from 192.168.192.4: icmp_seq=0 ttl=253 time=1.092 ms
64 bytes from 192.168.192.4: icmp_seq=1 ttl=253 time=1.019 ms
64 bytes from 192.168.192.4: icmp_seq=2 ttl=253 time=1.031 ms

--- 192.168.192.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.019/1.047/1.092/0.032 ms
```

To determine the path between these two interfaces, use the following traceroute command:

```
user@vpn5> traceroute 192.168.192.4 source 10.255.10.5
traceroute to 192.168.192.4 (192.168.192.4) from 10.255.10.5, 30 hops max, 40
byte packets
 1  vpn-08-t3003.isp-core.net (192.168.193.2)  0.678 ms  0.549 ms  0.494 ms
 2  vpn1-so-100.isp-core.net (192.168.192.100)  0.873 ms  0.847 ms  0.844 ms
    MPLS Label=100021 CoS=0 TTL=1 S=1
 3  vpn4-fe-112.isp-core.net (192.168.192.4)  0.825 ms  0.743 ms  0.764 ms
```

Ping the Directly Connected PE and CE Routers from Each Other

From the loopback interfaces on the CE routers, you can ping the VPN interface on the directly connected PE router. See Figure 38 for the topology referenced in these examples.

From the loopback interface on Router CE1 (VPN4), ping the VPN interface, fe-1/1/0.0, on Router PE1:

```
user@vpn4> ping 192.168.192.1 local 10.255.10.4 count 3
PING 192.168.192.1 (192.168.192.1): 56 data bytes
64 bytes from 192.168.192.1: icmp_seq=0 ttl=255 time=0.885 ms
64 bytes from 192.168.192.1: icmp_seq=1 ttl=255 time=0.757 ms
64 bytes from 192.168.192.1: icmp_seq=2 ttl=255 time=0.734 ms

--- 192.168.192.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.734/0.792/0.885/0.066 ms
```

To determine the path from the loopback interface on Router CE1 to the VPN interfaces on Router PE1, use the following traceroute command:

```
user@vpn4> traceroute 192.168.192.1 source 10.255.10.4
traceroute to 192.168.192.1 (192.168.192.1) from 10.255.10.4, 30 hops max, 40
byte packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.828 ms  0.657 ms  1.972 ms
```

From the loopback interface on Router CE2 (VPN5), ping the VPN interface, t3-0/0/3.0, on Router PE2:

```
user@vpn5> ping 192.168.193.2 local 10.255.10.5 count 3
PING 192.168.193.2 (192.168.193.2): 56 data bytes
64 bytes from 192.168.193.2: icmp_seq=0 ttl=255 time=0.998 ms
64 bytes from 192.168.193.2: icmp_seq=1 ttl=255 time=0.834 ms
64 bytes from 192.168.193.2: icmp_seq=2 ttl=255 time=0.819 ms

--- 192.168.193.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.819/0.884/0.998/0.081 ms
```

To determine the path from the loopback interface on Router CE2 to the VPN interfaces on Router PE2, use the following traceroute command:

```
serpil@vpn1> traceroute 192.168.193.2 source 10.255.10.5
traceroute to 192.168.193.2 (192.168.193.2) from 10.255.10.5, 30 hops max, 40
byte packets
 1  vpn-08-t3003.isp-core.net (192.168.193.2)  0.852 ms  0.670 ms  0.656 ms
```

From the VPN interface on the PE router, you can ping the VPN or loopback interface on the directly connected CE router.

From the VPN interface on Router PE1 (VPN1), ping the VPN interface on Router CE1, fe-1/1/0.0:

```
user@vpn1> ping 192.168.192.4 vpn-interface fe-1/1/0.0 local 192.168.192.1 count
3
PING 192.168.192.4 (192.168.192.4): 56 data bytes
64 bytes from 192.168.192.4: icmp_seq=0 ttl=255 time=0.866 ms
64 bytes from 192.168.192.4: icmp_seq=1 ttl=255 time=0.728 ms
64 bytes from 192.168.192.4: icmp_seq=2 ttl=255 time=0.753 ms

--- 192.168.192.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.728/0.782/0.866/0.060 ms
```

From the VPN interface on Router PE1 (VPN1), ping the loopback interface on Router CE1, 10.255.10.4:

```
user@vpn1> ping 10.255.10.4 vpn-interface fe-1/1/0.0 local 192.168.192.1 count 3
PING 10.255.10.4 (10.255.10.4): 56 data bytes
64 bytes from 10.255.10.4: icmp_seq=0 ttl=255 time=0.838 ms
64 bytes from 10.255.10.4: icmp_seq=1 ttl=255 time=0.760 ms
64 bytes from 10.255.10.4: icmp_seq=2 ttl=255 time=0.771 ms

--- 10.255.10.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.760/0.790/0.838/0.034 ms
```

To determine the path from the VPN interface on Router PE1 to the VPN and loopback interfaces on Router CE1, respectively, use the following traceroute commands:

```
user@vpn1> traceroute 10.255.10.4 vpn-interface fe-1/1/0.0 source 192.168.192.1
traceroute to 10.255.10.4 (10.255.10.4) from 192.168.192.1, 30 hops max, 40 byte
packets
 1  vpn4.isp-core.net (10.255.10.4)  0.842 ms  0.659 ms  0.621 ms

user@vpn1> traceroute 192.168.192.4 vpn-interface fe-1/1/0.0 source
192.168.192.1
traceroute to 192.168.192.4 (192.168.192.4) from 192.168.192.1, 30 hops max, 40
byte packets
 1  vpn4-fe-112.isp-core.net (192.168.192.4)  0.810 ms  0.662 ms  0.640 ms
```

From the VPN interface on Router PE2 (VPN2), ping the VPN interface on Router CE2, t3-0/0/3.0:

```
user@vpn2> ping 192.168.193.5 vpn-interface t3-0/0/3.0 local 192.168.193.2 count 3
PING 192.168.193.5 (192.168.193.5): 56 data bytes
64 bytes from 192.168.193.5: icmp_seq=0 ttl=255 time=0.852 ms
64 bytes from 192.168.193.5: icmp_seq=1 ttl=255 time=0.909 ms
64 bytes from 192.168.193.5: icmp_seq=2 ttl=255 time=0.793 ms

--- 192.168.193.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.793/0.851/0.909/0.047 ms
```

From the VPN interface on Router PE2 (VPN2), ping the loopback interface on Router CE2, 10.255.10.5:

```
user@vpn2> ping 10.255.10.5 vpn-interface t3-0/0/3.0 local 192.168.193.2 count 3
PING 10.255.10.5 (10.255.10.5): 56 data bytes
64 bytes from 10.255.10.5: icmp_seq=0 ttl=255 time=0.914 ms
64 bytes from 10.255.10.5: icmp_seq=1 ttl=255 time=0.888 ms
64 bytes from 10.255.10.5: icmp_seq=2 ttl=255 time=1.066 ms

--- 10.255.10.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.888/0.956/1.066/0.079 ms
```

To determine the path from the VPN interface on Router PE2 to the VPN and loopback interfaces on Router CE2, respectively, use the following traceroute commands:

```
user@vpn2> traceroute 10.255.10.5 vpn-interface t3-0/0/3.0 source 192.168.193.2
traceroute to 10.255.10.5 (10.255.10.5) from 192.168.193.2, 30 hops max, 40 byte packets
 1  vpn5.isp-core.net (10.255.10.5)  1.009 ms  0.677 ms  0.633 ms

user@vpn2> traceroute 192.168.193.5 vpn-interface t3-0/0/3.0 source 192.168.193.2
traceroute to 192.168.193.5 (192.168.193.5) from 192.168.193.2, 30 hops max, 40 byte packets
 1  vpn5-t3-003.isp-core.net (192.168.193.5)  0.974 ms  0.665 ms  0.619 ms
```


Chapter 22

Layer 3 VPN Configuration Examples

This chapter provides the following examples of Layer 3 Virtual Private Networks (VPNs) configuration:

- Configure a Simple Full-Mesh VPN Topology on page 227
- Configure a Full-Mesh VPN Topology with Route Reflectors on page 242
- Configure a Hub-and-Spoke VPN Topology on page 242
- Configure an LDP-over-RSVP VPN Topology on page 257
- Configure an Application-Based Layer 3 VPN Topology on page 272



Note

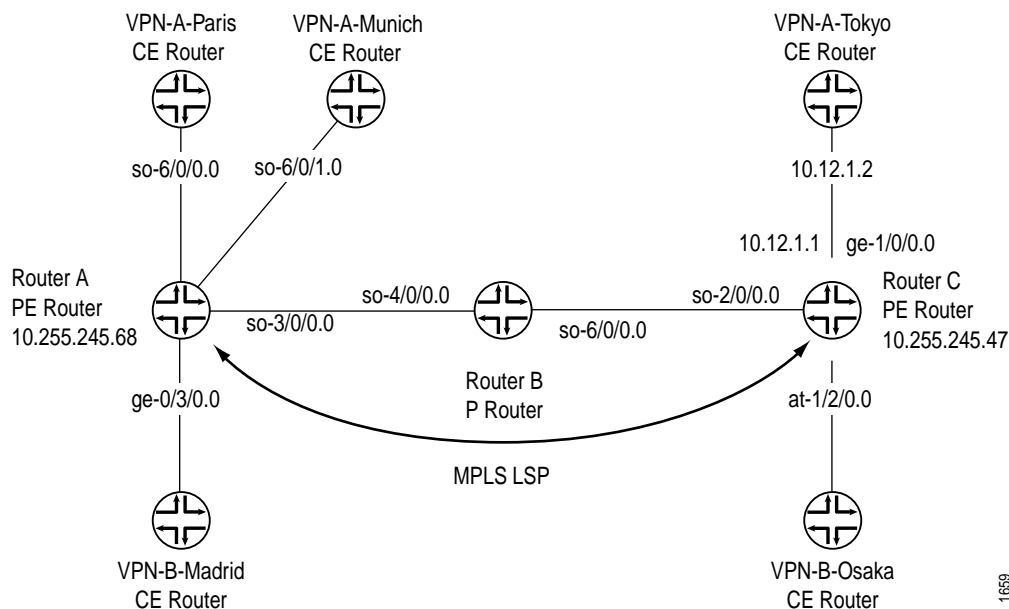
The examples in this chapter show only the portions of the configuration that establish VPN functionality. You must also configure other needed router functionality, including configuring all router interfaces, for a router configuration to work properly.

Configure a Simple Full-Mesh VPN Topology

This example shows how to set up a simple full-mesh service provider VPN configuration, which consists of the following components (see Figure 39):

- Two separate VPNs (VPN-A and VPN-B)
- Two provider edge (PE) routers, both of which service VPN-A and VPN-B
- RSVP as the signaling protocol
- One RSVP LSP that tunnels between the two PE routers through one provider (P) router

Figure 39: Example of a Simple VPN Topology



In this configuration, route distribution in VPN A from the router VPN-A-Paris to the router VPN-A-Tokyo occurs as follows:

1. The customer edge (CE) router VPN-A-Paris announces routes to the PE router Router A.
2. Router A installs the received announced routes into its VPN Routing and Forwarding (VRF) table, VPN-A.inet.0.
3. Router A creates an MPLS label for the interface between it and the router VPN-A-Paris.
4. Router A checks its VRF export policy.
5. Router A converts the IPv4 routes from VPN-A-Paris into VPN IPv4 format using its route distinguisher and announces these routes to PE Router C over the IBGP between the two PE routers.
6. Router C checks its VRF import policy and installs all routes that match the policy into its `bgp.l3vpn.0` routing table. (Any routes that do not match are discarded.)
7. Router C checks its VRF import policy and installs all routes that match into its VPN-A.inet.0 routing table. The routes are installed in IPv4 format.
8. Router C announces its routes to the CE router VPN-A-Tokyo, which installs them into its master routing table. (For routers running JUNOS software, the master routing table is `inet.0`.)
9. Router C uses the LSP between it and Router A to route all packets from router VPN-A-Tokyo that are destined for the router VPN-A-Paris.

The following sections explain how to configure the VPN functionality on the PE and provider routers. The CE routers are not aware of the VPN, so you configure them normally.

- Enable an IGP on the PE and Provider Routers on page 229
- Enable RSVP and MPLS on the Provider Router on page 229
- Configure the MPLS LSP Tunnel between the PE Routers on page 230
- Configure IBGP on the PE Routers on page 231
- Configure Routing Instances for VPNs on the PE Routers on page 232
- Configure VPN Policy on the PE Routers on page 234

The final section in this example, “Simple VPN Configuration Summarized by Router” on page 237, consolidates the statements needed to configure VPN functionality on each of the service provider routers shown in Figure 39.



Note

In this example, a private AS number is used for the route distinguisher and the route target. This number is used for illustrative purposes only. When you are configuring VPNs, you should use an assigned AS number.

Enable an IGP on the PE and Provider Routers

To allow the PE and provider routers to exchange routing information among themselves, you must configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the VPN routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

Enable RSVP and MPLS on the Provider Router

On the provider router, Router B, you must configure RSVP and MPLS because this router exists on the MPLS LSP path between the two PE routers, Router A and Router C:

```
[edit]
protocols {
  rsvp {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
  }
  mpls {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
  }
}
```

Configure the MPLS LSP Tunnel between the PE Routers

In this configuration example, RSVP is used for VPN signaling. Therefore, in addition to configuring RSVP, you must enable traffic engineering support in an IGP and you must create an MPLS LSP to tunnel the VPN traffic.

On PE Router A, enable RSVP and configure one end of the MPLS LSP tunnel. In this example, traffic engineering support is enabled for OSPF. When configuring the MPLS LSP, include interface statements for all interfaces participating in MPLS, including the interfaces to the PE and CE routers. The statements for the interfaces between the PE and CE routers are needed so that the PE router can create an MPLS label for the private interface. In this example, the first interface statement configures MPLS on the interface connected to the LSP, and the remaining three configure MPLS on the interfaces that connect the PE router to the CE routers.

```
[edit]
protocols {
  rsvp {
    interface so-3/0/0.0;
  }
  mpls {
    label-switched-path RouterA-to-RouterC {
      to 10.255.245.47;
    }
    interface so-3/0/0.0;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    interface ge-0/3/0.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-3/0/0.0;
    }
  }
}
```

On PE Router C, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and the CE routers.

```
[edit]
protocols {
  rsvp {
    interface so-2/0/0.0;
  }
  mpls {
    label-switched-path RouterC-to-RouterA {
      to 10.255.245.68;
    }
    interface so-2/0/0.0;
    interface ge-1/0/0.0;
    interface at-1/2/0.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-2/0/0.0;
    }
  }
}
```


Configure IBGP on the PE Routers

On the PE routers, configure an IBGP session with the following properties:

- VPN family—To indicate that the IBGP session is for the VPN, include the family inet-vpn statement.
- Loopback address—Include the local-address statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. Note that you must also configure the lo0 interface at the [edit interfaces] hierarchy level. The example does not include this part of the router's configuration.
- Neighbor address—Include the neighbor statement, specifying the IP address of the neighboring PE router, which is its loopback (lo0) address.

On PE Router A, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE-RouterA-to-PE-RouterC {
      type internal;
      local-address 10.255.245.68;
      family inet-vpn {
        unicast:
      }
      neighbor 10.255.245.47;
    }
  }
}
```

On PE Router C, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE-RouterC-to-PE-RouterA {
      type internal;
      local-address 10.255.245.47;
      family inet-vpn {
        unicast:
      }
      neighbor 10.255.245.68;
    }
  }
}
```

Configure Routing Instances for VPNs on the PE Routers

Both PE routers service VPN-A and VPN-B, so you must configure two routing instances on each router; one for each VPN. For each VPN, you must define the following in the routing instance:

- Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of vrf, which creates the VRF table on the PE router.
- Interfaces connected to the CE routers.
- VRF import and export policies, which must be the same on each PE router that services the same VPN. Unless an import policy contains only a then reject statement, it must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails.



Note

In this example, a private AS number is used for the route distinguisher. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

- Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—either BGP, OSPF, or RIP—or you can configure static routing.

On PE Router A, configure the following routing instance for VPN-A. In this example, Router A uses static routes to distribute routes to and from the two CE routers to which it is connected.

```
[edit]
routing-instance {
  VPN-A-Paris-Munich {
    instance-type vrf;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    route-distinguisher 65535:0;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    routing-options {
      static {
        route 172.16.0.0/16 next-hop so-0/0/0.0;
        route 172.17.0.0/16 next-hop so-6/0/1.0;
      }
    }
  }
}
```

On PE Router C, configure the following routing instance for VPN-A. In this example, Router C uses BGP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-A-Tokyo {
    instance-type vrf;
    interface ge-1/0/0.0;
    route-distinguisher 65535:1;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    protocols {
      bgp {
        group VPN-A-Site2 {
          peer-as 1;
          neighbor 10.12.1.2;
        }
      }
    }
  }
}
```

On PE Router A, configure the following routing instance for VPN-B. In this example, Router A uses OSPF to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-B-Madrid {
    instance-type vrf;
    interface ge-0/3/0.0;
    route-distinguisher 65535:2;
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
    protocols {
      ospf {
        area 0.0.0.0 {
          interface ge-0/3/0;
        }
      }
    }
  }
}
```

On PE Router C, configure the following routing instance for VPN-B. In this example, Router C uses RIP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-B-Osaka {
    instance-type vrf;
    interface at-1/2/0.0;
    route-distinguisher 65535:3;
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
    protocols {
      rip {
        group PE-C-to-VPN-B {
          neighbor at-1/2/0;
        }
      }
    }
  }
}
```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which they use to forward packets within a VPN. For VPN-A, the VRF table is VPN-A.inet.0, and for VPN-B it is VPN-B.inet.0.

In the VPN policy, you also configure VPN target communities.



Note

In this example, a private AS number is used for the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

On PE Router A, configure the following VPN import and export policies.



Note

The policy qualifiers shown in this example are only those needed for the VPN to function. You can configure additional qualifiers, as needed, to any policies that you configure.

```

[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol static;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-import {
    term a {
      from {
        protocol bgp;
        community VPN-B;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-export {
    term a {
      from protocol ospf;
      then {
        community add VPN-B;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:00;
  community VPN-B members target:65535:01;
}

```

On PE Router C, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol bgp;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-import {
    term a {
      from {
        protocol bgp;
        community VPN-B;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-export {
    term a {
      from protocol rip;
      then {
        community add VPN-B;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:00;
  community VPN-B members target:65535:01;
}
```

To apply the VPN policies on the routers, include the `vrf-export` and `vrf-import` statements when you configure the routing instance. For both VPNs, the VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

To apply the VPN policies on PE Router A, include the following statements:

```
[edit]
routing-instance {
  VPN-A-Paris-Munich {
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
  }
  VPN-B-Madrid {
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
  }
}
```

To apply the VPN policies on PE Router C, include the following statements:

```
[edit]
routing-instance {
  VPN-A-Tokyo {
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
  }
  VPN-B-Osaka {
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
  }
}
```

Simple VPN Configuration Summarized by Router

Router A (PE Router)

Routing Instance for VPN-A	<pre>routing-instance { VPN-A-Paris-Munich { instance-type vrf; interface so-6/0/0.0; interface so-6/0/1.0; route-distinguisher 65535:0; vrf-import VPN-A-import; vrf-export VPN-A-export; } }</pre>
Instance Routing Protocol	<pre>routing-options { static { route 172.16.0.0/16 next-hop so-6/0/0.0; route 172.17.0.0/16 next-hop so-6/0/1.0; } }</pre>
Routing Instance for VPN-B	<pre>routing-instance { VPN-B-Madrid { instance-type vrf; interface ge-0/3/0.0; route-distinguisher 65535:2; vrf-import VPN-B-import; vrf-export VPN-B-export; } }</pre>

```

Instance Routing Protocol      protocols {
                                ospf {
                                  area 0.0.0.0 {
                                    interface ge-0/3/0;
                                  }
                                }
                              }
Master Protocol Instance      protocols {

Enable RSVP                    rsvp {
                                interface so-3/0/0.0;
                              }

Configure an MPLS LSP          mpls {
                                label-switched-path RouterA-to-RouterC {
                                  to 10.255.245.47;
                                }
                                interface so-3/0/0.0;
                                interface so-6/0/0.0;
                                interface so-6/0/1.0;
                                interface ge-0/3/0.0;
                              }

Configure IBGP                  bgp {
                                group PE-RouterA-to-PE-RouterC {
                                  type internal;
                                  local-address 10.255.245.68;
                                  family inet-vpn {
                                    unicast;
                                  }
                                  neighbor 10.255.245.47;
                                }
                              }

Configure OSPF for Traffic     ospf {
  Engineering Support          traffic-engineering;
                                area 0.0.0.0 {
                                  interface so-3/0/0.0;
                                }
                              }

Configure VPN Policy            policy-options {
                                policy-statement VPN-A-import {
                                  term a {
                                    from {
                                      protocol bgp;
                                      community VPN-A;
                                    }
                                    then accept;
                                  }
                                  term b {
                                    then reject;
                                  }
                                }
                              }

```



```

policy-statement VPN-A-export {
  term a {
    from protocol static;
    then {
      community add VPN-A;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement VPN-B-import {
  term a {
    from {
      protocol bgp;
      community VPN-B;
    }
    then accept;
  }
  term b {
    then reject;
  }
}
policy-statement VPN-B-export {
  term a {
    from protocol ospf;
    then {
      community add VPN-B;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community VPN-A members target:65535:00;
community VPN-B members target:65535:01;
}

```

Router B (Provider Router)

```

Master Protocol Instance protocols {

  Enable RSVP      rsvp {
                    interface so-4/0/0.0;
                    interface so-6/0/0.0;
                    }

  Enable MPLS      mpls {
                    interface so-4/0/0.0;
                    interface so-6/0/0.0;
                    }
}

```

Router C (PE Router)

```

Routing Instance for VPN-A  routing-instance {
                               VPN-A-Tokyo {
                               instance-type vrf;
                               interface ge-1/0/0.0;
                               route-distinguisher 65535:1;
                               vrf-import VPN-A-import;
                               vrf-export VPN-A-export;

Instance Routing Protocol    protocols {
                               bgp {
                               group VPN-A-Site2 {
                               peer-as 1;
                               neighbor 10.12.1.2;
                               }
                               }
                               }
                               }

Routing Instance for VPN-B  VPN-B-Osaka {
                               instance-type vrf;
                               interface at-1/2/0.0;
                               route-distinguisher 65535:3;
                               vrf-import VPN-B-import;
                               vrf-export VPN-B-export;

Instance Routing Protocol    protocols {
                               rip {
                               group PE-C-to-VPN-B {
                               neighbor at-1/2/0;
                               }
                               }
                               }
                               }

Master Protocol Instance    protocols {

Enable RSVP                 rsvp {
                               interface so-2/0/0.0;
                               }

Configure an MPLS LSP       mpls {
                               label-switched-path RouterC-to-RouterA {
                               to 10.255.245.68;
                               }
                               interface so-2/0/0.0;
                               interface ge-1/0/0.0;
                               interface at-1/2/0.0;
                               }

Configure IBGP             bgp {
                               group PE-RouterC-to-PE-RouterA {
                               type internal;
                               local-address 10.255.245.47;
                               family inet-vpn {
                               unicast;
                               }
                               neighbor 10.255.245.68;
                               }
                               }
                               }

```

Configure OSPF for Traffic Engineering Support

```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface so-2/0/0.0;
  }
}
```

Configure VPN Policy

```
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol bgp;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-import {
    term a {
      from {
        protocol bgp;
        community VPN-B;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-export {
    term a {
      from protocol rip;
      then {
        community add VPN-B;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:00;
  community VPN-B members target:65535:01;
}
```

Configure a Full-Mesh VPN Topology with Route Reflectors

This example is a variation of the full-mesh VPN topology example (described in “Configure a Simple Full-Mesh VPN Topology” on page 227) in which one of the PE routers is a BGP route reflector. In this variation, Router C in Figure 39 on page 228 is a route reflector. The only change to its configuration is that you need to include the cluster statement when configuring the BGP group:

```
[edit protocols]
bgp {
  group PE-RouterC-to-PE-RouterA {
    type internal;
    local-address 10.255.245.47;
    family inet-vpn {
      unicast:
    }
    neighbor 10.255.245.68;
    cluster 4.3.2.1;
  }
}
```

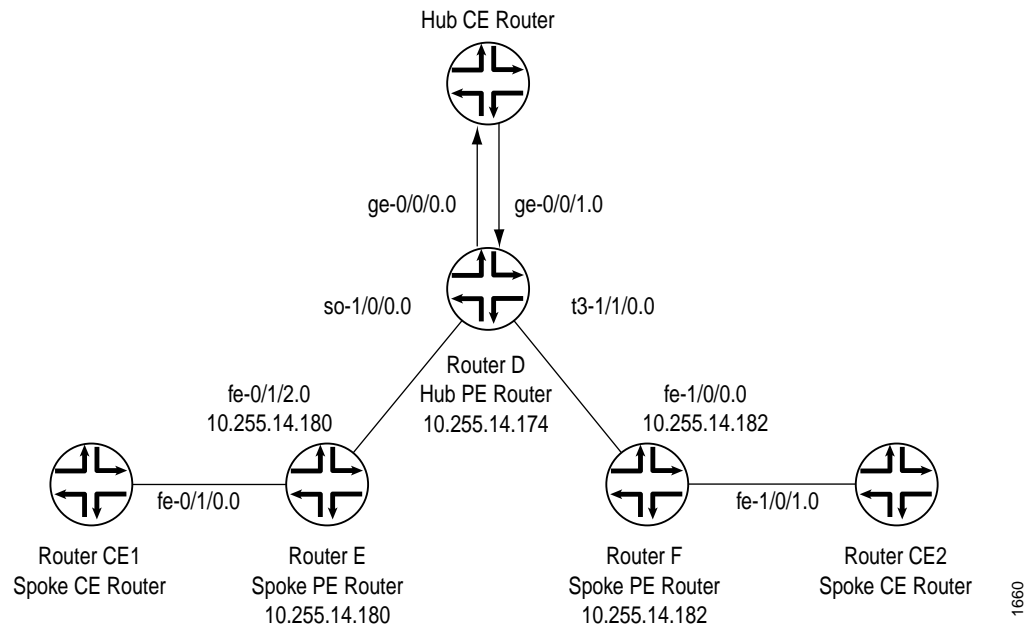
For the complete example of a configuration of Router C, see “Router C (PE Router)” in “Simple VPN Configuration Summarized by Router” on page 237.

Configure a Hub-and-Spoke VPN Topology

This example shows how to set up a hub-and-spoke VPN configuration, which consists of the following components (see Figure 40):

- One hub PE router (Router D).
- One hub CE router connected to the hub PE router. For a hub-and-spoke VPN topology to function properly, there must be two interfaces connecting the hub PE router to the hub CE router, and each interface must have its own VRF table on the PE router:
 - One interface (here, interface ge-0/0/0.0) is used to announce spoke routes to the hub CE router. The VRF table associated with this interface contains the routes being announced by the spoke PE routers to the hub CE router.
 - The second interface (here, interface ge-0/0/1.0) is used to receive route announcements from the hub CE that are destined for the hub and spoke routers. The VRF table associated with this interface contains the routes announced by the hub CE router to the spoke PE routers.
- Two spoke PE routers (Router E and Router F).
- Two spoke CE routers (CE1 and CE2), one connected to each spoke PE router.
- LDP as the signaling protocol.

Figure 40: Example of a Hub-and-Spoke VPN Topology

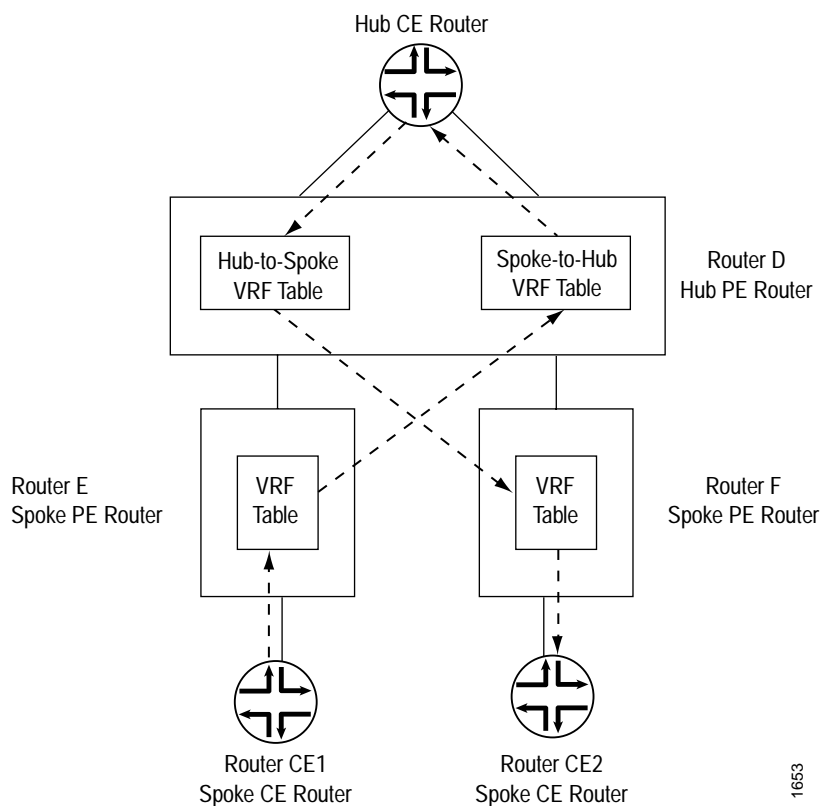


In this configuration, route distribution from spoke CE Router CE1 occurs as follows:

1. Spoke Router CE1 announces its routes to spoke PE Router E.
2. Router E installs the routes from CE1 into its VRF table.
3. After checking its VRF export policy, Router E adds the spoke target community to the routes from Router CE1 that passed the policy and announces them to the hub PE router, Router D.
4. Router D checks the VRF import policy associated with interface `ge-0/0/0.0` and places all routes from spoke PE routers that match the policy into its `bgp.l3vpn` routing table. (Any routes that do not match are discarded.)
5. Router D checks its VRF import policy associated with interface `ge-0/0/0.0` and installs all routes that match into its spoke VRF table. The routes are installed with the spoke target community.
6. Router D announces routes to the hub CE over interface `ge-0/0/0.0`.
7. The hub CE router announces the routes back to the hub PE Router D over the second interface to the hub router, interface `ge-0/0/1.0`.
8. The hub PE installs the routes learned from the hub CE router into its hub VRF table, which is associated with interface `ge-0/0/1.0`.
9. The hub PE checks the VRF export policy associated with interface `ge-0/0/1.0` and announces all routes that match to all spokes after adding the hub target community.

Figure 41 illustrates how routes are distributed from this spoke router to the other spoke CE router, Router CE2. The path shown is also the same one that is followed if you issue a traceroute command from Router CE1 to Router CE2.

Figure 41: Route Distribution between Two Spoke Routers



1653

The following sections explain how to configure the VPN functionality for a hub-and-spoke topology on the hub and spoke PE routers. The CE routers do not know about the VPN, so you configure them normally.

- Enable an IGP on the Hub and Spoke PE Routers on page 245
- Configure LDP on the Hub and Spoke PE Routers on page 245
- Configure IBGP on the PE Routers on page 246
- Configure Routing Instances for VPNs on the Hub and Spoke PE Routers on page 247
- Configure VPN Policy on the PE Routers on page 249

The final section in this example, “Hub-and-Spoke VPN Configuration Summarized by Router” on page 252, consolidates the statements needed to configure VPN functionality for each of the service provider routers shown in Figure 39.

Enable an IGP on the Hub and Spoke PE Routers

To allow the hub and spoke PE routers to exchange routing information, you must configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

In the route distribution in a hub-and-spoke topology, if the protocol used between the CE and PE routers at the hub site is BGP, the hub CE router announces all routes received from the hub PE router and the spoke routers back to the hub PE router and all the spoke routers. This means that the hub and spoke PE routers receive routes that contain their AS number. Normally, when a route contains this information, it indicates that a routing loop has occurred and the router rejects the routes. However, for the VPN configuration to work, the hub PE router and the spoke routers must accept these routes. To enable this, include the loops option when configuring the AS at the [edit routing-options] hierarchy level on the hub PE router and all the spoke routers. For this example configuration, you specify a value of 1. You can specify a number from 0 through 10.

```
[edit routing-options]
autonomous-system as-number loops 1;
```

Configure LDP on the Hub and Spoke PE Routers

You must configure LDP on the interfaces between the hub and spoke PE routers that participate in the VPN.

On hub PE Router D, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-1/0/0.0;
  interface t3-1/1/0.0;
}
```

On spoke PE Router E, configure LDP as follows:

```
[edit protocols]
ldp {
  interface fe-0/1/2.0;
}
```

On spoke PE router F, configure LDP as follows:

```
[edit protocols]
ldp {
  interface fe-1/0/0.0;
}
```

Configure IBGP on the PE Routers

On the hub and spoke PE routers, configure an IBGP session with the following properties:

- **VPN family**—To indicate that the IBGP session is for the VPN, include the family inet-vpn statement.
- **Loopback address**—Include the local-address statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. Note that you must also configure the lo0 interface at the [edit interfaces] hierarchy level. The example does not include this part of the router's configuration.
- **Neighbor address**—Include the neighbor statement. On the hub router, specify the IP address of each spoke PE router, and on the spoke router, specify the address of the hub PE router.

For the hub router, you configure an IBGP session with each spoke, and for each spoke router, you configure an IBGP session with the hub. There are no IBGP sessions between the two spoke routers.

On hub Router D, configure IBGP as follows. The first neighbor statement configures an IBGP session to spoke Router E, and the second configures a session to spoke Router F.

```
[edit protocols]
bgp {
  group Hub-to-Spokes {
    type internal;
    local-address 10.255.14.174;
    family inet-vpn {
      unicast;
    }
    neighbor 10.255.14.180;
    neighbor 10.255.14.182;
  }
}
```

On spoke Router E, configure an IBGP session to the hub router as follows:

```
[edit protocols]
bgp {
  group Spoke-E-to-Hub {
    type internal;
    local-address 10.255.14.180;
    neighbor 10.255.14.174 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```


On spoke Router F, configure an IBGP session to the hub router as follows:

```
[edit protocols]
bgp {
  group Spoke-F-to-Hub {
    type internal;
    local-address 10.255.14.182;
    neighbor 10.255.14.174 {
      family inet-vpn {
        unicast:
      }
    }
  }
}
```

Configure Routing Instances for VPNs on the Hub and Spoke PE Routers

For the hub PE router to be able to distinguish between packets going to and coming from the spoke PE routers, you must configure it with two routing instances:

- One routing instance (in this example, Spokes-to-Hub-CE) is associated with the interface that carries packets from the hub PE router to the hub CE router (in this example, interface ge-0/0/0.0). Its VRF table contains the routes being announced by the spoke PE routers and the hub PE router to the hub CE router.
- The second routing instance (in this example, Hub-CE-to-Spokes) is associated with the interface that carries packets from the hub CE router to the hub PE router (in this example, interface ge-0/0/1.0). Its VRF table contains the routes being announced from the hub CE router to the hub and spoke PE routers.

On each spoke router, you must configure one routing instance.

You must define the following in the routing instance:

- Route distinguisher, which is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of `vrf`, which creates the VRF table on the PE router.
- Interfaces that are part of the VPN and that connect the PE routers to their CE routers.
- VRF import and export policies. Both import policies must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails. (The exception to this is if the import policy contains only a `then reject` statement.) In the VRF export policy, spoke PE routers attach the spoke target community.
- Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—either BGP, OSPF, or RIP—or you can configure static routing.

For a hub-and-spoke topology, you must configure different policies in each routing instance on the hub CE router. For the routing instance associated with the interface that carries packets from the hub PE router to the hub CE router (in this example, Spokes-to-Hub-CE), the import policy must accept all routes received on the IBGP session between the hub and spoke PE routers and the export policy must reject all routes received from the hub CE router. For the routing instance associated with the interfaces that carries packets from the hub CE router to the hub PE router (in this example, Hub-CE-to-Spokes), the import policy must reject all routes received from the spoke PE routers, and the export policy must export to all the spoke routers.

On hub PE Router D, configure the following routing instances. Router D uses OSPF to distribute routes to and from the hub CE router.

```
[edit]
routing-instance {
  Spokes-to-Hub-CE {
    instance-type vrf;
    interface ge-0/0/0.0;
    route-distinguisher 10.255.1.174:65535;
    vrf-import spoke;
    vrf-export null;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface ge-0/0/0;
        }
      }
    }
  }
  Hub-CE-to-Spokes {
    instance-type vrf;
    interface ge-0/0/1.0;
    route-distinguisher 10.255.1.174:65535;
    vrf-import null;
    vrf-export hub;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface ge-0/0/1.0;
        }
      }
    }
  }
}
```

On spoke PE Router E, configure the following routing instances. Router E uses OSPF to distribute routes to and from the spoke CE router CE1.

```
[edit]
routing-instance {
  Spoke-E-to-Hub {
    instance-type vrf;
    interface fe-0/1/0.0;
    route-distinguisher 10.255.14.80:65535;
    vrf-import hub;
    vrf-export spoke;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface fe-0/1/0.0;
        }
      }
    }
  }
}
```

On spoke PE Router F, configure the following routing instances. Router F uses OSPF to distribute routes to and from the spoke CE router CE2.

```
[edit]
routing-instance {
  Spoke-F-to-Hub {
    instance-type vrf;
    interface fe-1/0/1.0;
    route-distinguisher 10.255.14.182:65535;
    vrf-import hub;
    vrf-export spoke;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface fe-1/0/1.0;
        }
      }
    }
  }
}
```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the hub and spoke PE routers so that they install the appropriate routes in the VRF tables, which they use to forward packets within each VPN.

On the spoke routers, you define policies to exchange routes with the hub router.

On the hub router, you define policies to accept routes from the spoke PE routers and distribute them to the hub CE router, and vice versa. The hub PE router has two VRF tables:

- Spoke-to-hub VRF table—Handles routes received from spoke routers and announces these routes to the hub CE router. For this VRF table, the import policy must check that the spoke target name is present and that the route was received from the IBGP session between the hub PE and the spoke PE routers. This VRF table must not export any routes, so its export policy should reject everything.
- Hub-to-spoke VRF table—Handles routes received from the hub CE router and announces them to the spoke routers. For this VRF table, the export policy must add the hub target community. This VRF table must not import any routes, so its import policy should reject everything.

In the VPN policy, you also configure the VPN target communities.

On hub PE Router D, configure the following policies to apply to the VRF tables:

- spoke—Accepts routes received from the IBGP session between it and the spoke PE routers that contain the community target spoke, and rejects all other routes.
- hub—Adds the community target hub to all routes received from OSPF (that is, from the session between it and the hub CE router). It rejects all other routes.
- null—Rejects all routes.
- redistribute-vpn—Redistributes OSPF routes to neighbors within the routing instance.

```
[edit]
policy-options {
  policy-statement spoke {
    term a {
      from {
        protocol bgp;
        community spoke;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement hub {
    term a {
      from protocol ospf;
      then {
        community add hub;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
}
```

```

policy-statement null {
  then reject;
}
policy-statement redistribute-vpn {
  term a {
    from protocol bgp;
    then accept;
  }
  term b {
    then reject;
  }
}
community hub members target:65535:1;
community spoke members target:65535:2;
}

```

To apply the VRF policies on Router D, include the vrf-export and vrf-import statements when you configure the routing instances:

```

[edit]
routing-instance {
  Spokes-to-Hub-CE {
    vrf-import spoke;
    vrf-export null;
  }
  Hub-CE-to-Spokes {
    vrf-import null;
    vrf-export hub;
  }
}

```

On spoke PE Router E and Router F, configure the following policies to apply to the VRF tables:

- **hub**—Accepts routes received from the IBGP session between it and the hub PE routers that contain the community target hub, and rejects all other routes.
- **spoke**—Adds the community target spoke to all routes received from OSPF (that is, from the session between it and the hub CE router) and rejects all other routes.
- **redistribute-vpn**—Redistributes OSPF routes to neighbors within the routing instance.

On spoke PE Router E and Router F, configure the following VPN import and export policies:

```

[edit]
policy-options {
  policy-statement hub {
    term a {
      from {
        protocol bgp;
        community hub;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}

```

```

policy-statement spoke {
  term a {
    from protocol ospf;
    then {
      community add spoke;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement redistribute-vpn {
  term a {
    from protocol bgp;
    then accept;
  }
  term b {
    then reject;
  }
}
community hub members target:65535:1;
community spoke members target 65535:2;
}

```

To apply the VRF policies on the spoke routers, include the vrf-export and vrf-import statements when you configure the routing instances:

```

[edit]
routing-instance {
  Spoke-E-to-Hub {
    vrf-import hub;
    vrf-export spoke;
  }
}

[edit]
routing-instance {
  Spoke-F-to-Hub {
    vrf-import hub;
    vrf-export spoke;
  }
}

```

Hub-and-Spoke VPN Configuration Summarized by Router

Router D (Hub PE Router)

Routing Instance for Distributing Spoke Routes to Hub CE	<pre> routing-instance { Spokes-to-Hub-CE { instance-type vrf; interface ge-0/0/0.0; route-distinguisher 10.255.1.174:65535; vrf-import spoke; vrf-export null; } } </pre>
---	--

Instance Routing Protocol	<pre> protocols { ospf { export redistribute-vpn; area 0.0.0.0 { interface ge-0/0/0; } } } </pre>
Routing Instance for Distributing Hub CE Routes to Spokes	<pre> Hub-CE-to-Spokes { instance-type vrf; interface ge-0/0/1.0; route-distinguisher 10.255.1.174:65535; vrf-import null; vrf-export hub; } </pre>
Instance Routing Protocols	<pre> protocols { ospf { export redistribute-vpn; area 0.0.0.0 { interface ge-0/0/1.0; } } } </pre>
Routing Options (Master Instance)	<pre> routing-options { autonomous-system 1 loops 1; } </pre>
Protocols (Master Instance)	<pre> protocols { </pre>
Enable LDP	<pre> ldp { interface so-1/0/0.0; interface t3-1/1/0.0; } </pre>
Configure IBGP	<pre> bgp { group Hub-to-Spokes { type internal; local-address 10.255.14.174; family inet-vpn { unicast: } neighbor 10.255.14.180; neighbor 10.255.14.182; } } } </pre>

```

Configure VPN Policy policy-options {
    policy-statement spoke {
        term a {
            from {
                protocol bgp;
                community spoke;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement hub {
        term a {
            from protocol ospf;
            then {
                community add hub;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    policy-statement null {
        then reject;
    }
    policy-statement redistribute-vpn {
        term a {
            from protocol bgp;
            then accept;
        }
        term b {
            then reject;
        }
    }
    community hub members target:65535:1;
    community spoke members target:65535:2;
}

```

Router E (Spoke PE Router)

```

Routing Instance routing-instance {
    Spoke-E-to-Hub {
        instance-type vrf;
        interface fe-0/1/0.0;
        route-distinguisher 10.255.14.80:65535;
        vrf-import hub;
        vrf-export spoke;
    }
}

```

```

Instance Routing Protocol protocols {
    ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
            interface fe-0/1/0.0;
        }
    }
}
}

```



```

Routing Options      routing-options {
(Master Instance)    autonomous-system 1 loops 1;
                        }

                        Protocols protocols {
(Master Instance)

                        Enable LDP    ldp {
                                interface fe-0/1/2.0;
                                }

                        Configure IBGP  bgp {
                                group Spoke-E-to-Hub {
                                    type internal;
                                    local-address 10.255.14.180;
                                    neighbor 10.255.14.174 {
                                        family inet-vpn {
                                            unicast:
                                        }
                                    }
                                }
                            }
                        }

Configure VPN Policy policy-options {
                        policy-statement hub {
                            term a {
                                from {
                                    protocol bgp;
                                    community hub;
                                }
                                then accept;
                            }
                            term b {
                                then reject;
                            }
                        }
                        policy-statement spoke {
                            term a {
                                from protocol ospf;
                                then {
                                    community add spoke;
                                    accept;
                                }
                            }
                            term b {
                                then reject;
                            }
                        }
                        policy-statement redistribute-vpn {
                            term a {
                                from protocol bgp;
                                then accept;
                            }
                            term b {
                                then reject;
                            }
                        }
                        community hub members target:65535:1;
                        community spoke members target:65535:2;
                    }

```

Router F (Spoke PE Router)

```

Routing Instance  routing-instance {
                   Spoke-F-to-Hub {
                     instance-type vrf;
                     interface fe-1/0/1.0;
                     route-distinguisher 10.255.14.182:65535;
                     vrf-import hub;
                     vrf-export spoke;

```

```

Instance Routing Protocol  protocols {
                           ospf {
                             export redistribute-vpn;
                             area 0.0.0.0 {
                               interface fe-1/0/1.0;
                             }
                           }
                         }

```

```

Routing Options  routing-options {
(Master Instance)  autonomous-system 1 loops 1;
                   }

```

```

Protocols  protocols {
(Master Instance)

```

```

Enable LDP  ldp {
            interface fe-1/0/0.0;
            }

```

```

Configure IBGP  bgp {
                group Spoke-F-to-Hub {
                  type internal;
                  local-address 10.255.14.182;
                  neighbor 10.255.14.174 {
                    family inet-vpn {
                      unicast;
                    }
                  }
                }
              }

```

```

Configure VPN Policy  policy-options {
                        policy-statement hub {
                          term a {
                            from {
                              protocol bgp;
                              community hub;
                            }
                            then accept;
                          }
                          term b {
                            then reject;
                          }
                        }
                        policy-statement spoke {
                          term a {
                            from protocol ospf;
                            then {
                              community add spoke;
                              accept;
                            }
                          }
                          term b {
                            then reject;
                          }
                        }
                        policy-statement redistribute-vpn {
                          term a {
                            from {
                              protocol bgp;
                            }
                            then accept;
                          }
                          term b {
                            then reject;
                          }
                        }
                        community hub members target:65535:1;
                        community spoke members target:65535:2;
                      }

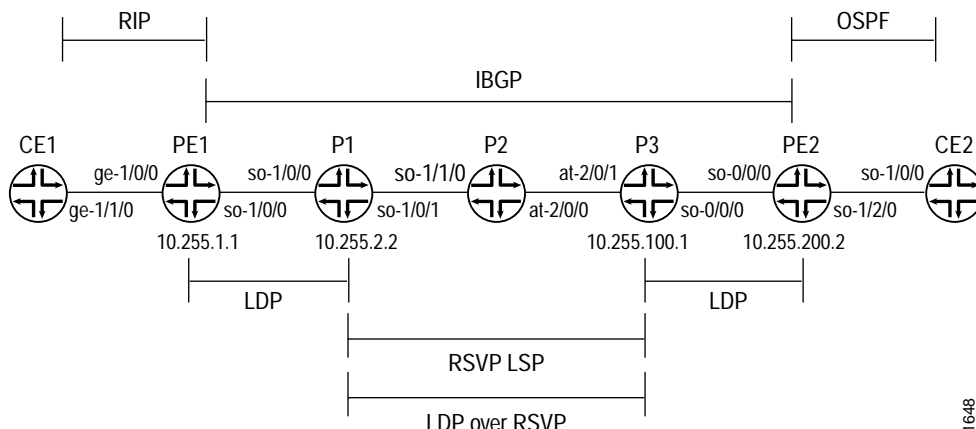
```

Configure an LDP-over-RSVP VPN Topology

This example shows how to set up a VPN topology in which LDP packets are tunneled over an RSVP LSP. This configuration consists of the following components (see Figure 42):

- One VPN (VPN-A)
- Two PE routers
- LDP as the signaling protocol between the PE routers and their adjacent provider routers
- An RSVP LSP between two of the provider routers over which LDP is tunneled

Figure 42: Example of an LDP-over-RSVP VPN Topology



The following steps describe how this topology is established and how packets are sent from CE Router CE2 to CE Router CE1:

1. The provider routers P1 and P3 establish RSVP LSPs between each other and install their loopback addresses in their inet.3 routing tables.
2. PE Router PE1 establishes an LDP session with Router P1 over interface so-1/0/0.0.
3. Router P1 establishes an LDP session with Router P3's loopback address, which is reachable using the RSVP LSP.
4. Router P1 sends its label bindings, which include a label to reach Router PE1, to Router P3. These label bindings allow Router P3 to direct LDP packets to Router PE1.
5. Router P3 establishes an LDP session with Router PE2 over interface so-0/0/0.0 and establishes an LDP session with Router P1's loopback address.
6. Router P3 sends its label bindings, which include a label to reach Router PE2, to Router P1. These label bindings allow Router P1 to direct LDP packets to Router PE2's loopback address.
7. Routers PE1 and PE2 establish IBGP sessions with each other.
8. When Router PE1 announces to Router PE2 routes that it learned from Router CE1, it includes its VPN label. (The PE router creates the VPN label and binds it to the interface between the PE and CE routers.) Similarly, when Router PE2 announces routes that it learned from Router CE2, it sends its VPN label to Router PE1.

When Router PE2 wants to forward a packet to Router CE1, it pushes two labels onto the packet's label stack: first, the VPN label that is bound to the interface between Router PE1 and Router CE1, then the LDP label used to reach Router PE1. Then, it forwards the packets to Router P3 over interface so-0/0/1.0.

9. When Router P3 receives the packets from Router PE2, it swaps the LDP label that is on top of the stack (according to its LDP database) and also pushes an RSVP label onto the top of the stack so that the packet can now be switched by the RSVP LSP. At this point, there are three labels on the stack: the inner (bottom) label is the VPN label, the middle is the LDP label, and the outer (top) is the RSVP label.

16-48

10. Router P2 receives the packet and switches it to Router P1 by swapping the RSVP label. In this topology, because Router P2 is the penultimate-hop router in the LSP, it pops the RSVP label and forwards the packet over interface so-1/1/0.0 to Router P1. At this point, there are two labels on the stack: the inner label is the VPN label and the outer one is the LDP label.
11. When Router P1 receives the packet, it pops the outer label (the LDP label) and forwards the packet to Router PE1 using interface so-1/0/0.0. In this topology, Router PE1 is the egress LDP router, so Router P1 pops the LDP label instead of swapping it with another label. At this point, there is only one label on the stack, the VPN label.
12. When Router PE1 receives the packet, it pops the VPN label and forwards the packet as an IPv4 packet to Router CE1 over interface ge-1/1/0.0.

A similar set of operations occurs for packets sent from Router CE1 that are destined for Router CE2.

The following list explains how, for packets being sent from Router CE2 to Router CE1, the LDP, RSVP, and VPN labels are announced by the various routers. These steps include examples of label values (illustrated in Figure 43).

■ LDP labels

- Router PE1 announces LDP label 3 for itself to Router P1.
- Router P1 announces LDP label 100,001 for Router PE1 to Router P3.
- Router P3 announces LDP label 100,002 for Router PE1 to Router PE2.

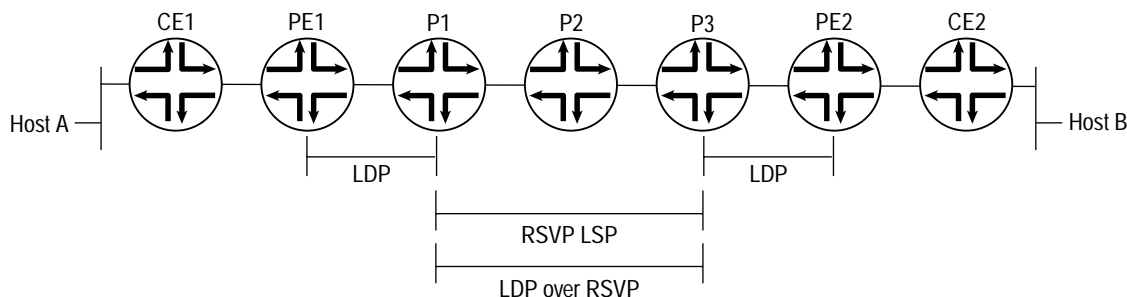
■ RSVP labels

- Router P1 announces RSVP label 3 to Router P2.
- Router P2 announces RSVP label 100,003 to Router P3.

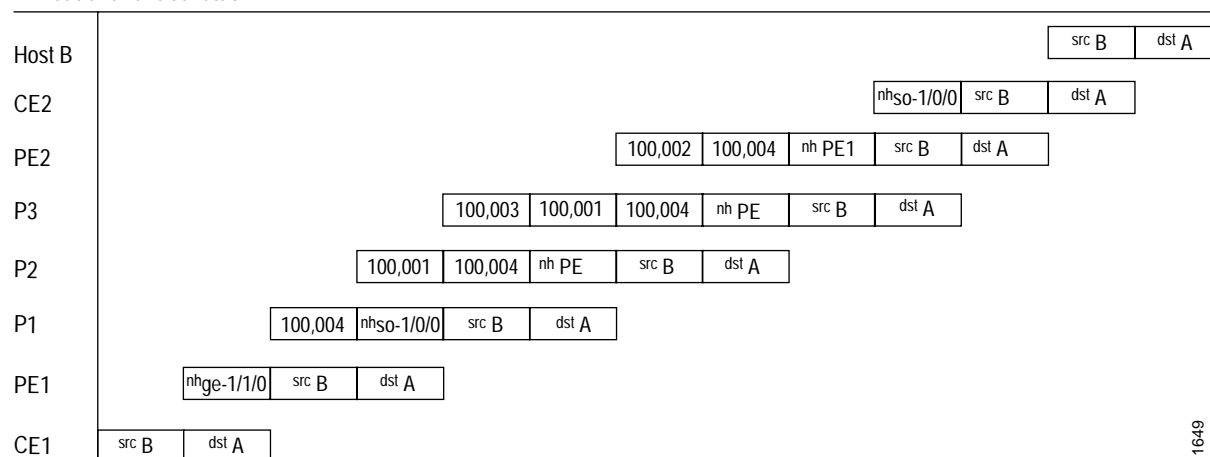
■ VPN label

- Router PE1 announces VPN label 100,004 to Router PE2 for the route from Router CE1 to Router CE2.

Figure 43: Label Pushing and Popping



IP header and label stack



1649

For a packet being sent from Host B in Figure 43 to Host A, the packet headers and labels change as follows as the packet travels to its destination:

1. The packet that originates from Host B has a source address of B and a destination address of A in its header.
2. Router CE2 adds to the packet a next hop of interface so-1/0/0.
3. Router PE2 swaps out the next hop of interface so-1/0/0 and replaces it with a next hop of PE1. It also adds two labels for reaching Router PE1, first the VPN label (100,004), then the LDP label (100,002). The VPN label is thus the inner (bottom) label on the stack, and the LDP label is the outer label.
4. Router P3 swaps out the LDP label added by Router PE2 (100,002) and replaces it with its LDP label for reaching Router PE1 (100,001). It also adds the RSVP label for reaching Router P2 (100,003).
5. Router P2 removes the RSVP label (100,003) because it is the penultimate hop in the MPLS LSP.
6. Router P1 removes the LDP label (100,001) because it is the penultimate LDP router. It also swaps out the next hop of PE1 and replaces it with the next hop interface, so-1/0/0.

7. Router PE1 removes the VPN label (100,004). It also swaps out the next hop interface of so-1/0/0 and replaces it with its next hop interface, ge-1/1/0.
8. Router CE1 removes the next hop interface of ge-1/1/0, and the packet header now contains just a source address of B and a destination address of A.

The following sections explain how to configure the VPN functionality on the PE and provider routers. The CE routers are not aware of the VPN, so you configure them normally.

- Enable an IGP on the PE and Provider Routers on page 261
- Enable LDP on the PE and Provider Routers on page 261
- Enable RSVP and MPLS on the Provider Router on page 263
- Configure the MPLS LSP Tunnel between the Provider Routers on page 263
- Configure IBGP on the PE Routers on page 264
- Configure Routing Instances for VPNs on the PE Routers on page 265
- Configure VPN Policy on the PE Routers on page 266

The final section in this example, “LDP-over-MPLS VPN Configuration Summarized by Router” on page 268, consolidates the statements needed to configure VPN functionality on each of the service provider routers shown in Figure 42.



Note

In this example, a private AS number is used for the route distinguisher and the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

Enable an IGP on the PE and Provider Routers

To allow the PE and provider routers to exchange routing information among themselves, you must configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the VPN routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

Enable LDP on the PE and Provider Routers

In this configuration example, LDP is the signaling protocol between the PE routers. For the VPN to function, you must configure LDP on the two PE routers and on the providers routers that are connected to the PE routers. You need to configure LDP only on the interfaces in the core of the service provider’s network, that is, between the PE and provider routers and between the provider routers. You do not need to configure LDP on the interface between the PE and CE routers.

In this configuration example, you configure LDP on the provider routers' loopback interfaces because these are the interfaces on which the MPLS LSP is configured.

On the PE routers, you must also configure family inet when you configure the logical interface.

On Router PE1, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-1/0/0.0;
}
[edit interfaces]
so-1/0/0 {
  unit 0 {
    family mpls;
  }
}
```

On Router PE2, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-0/0/0.0;
}
[edit interfaces]
so-0/0/1 {
  unit 0 {
    family mpls;
  }
}
```

On Router P1, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-1/0/0.0;
  interface lo0;
}
```

On Router P3, configure LDP as follows:

```
[edit protocols]
ldp {
  interface lo0;
  interface so-0/0/0.0;
}
```

On Router P2, while you do not need to configure LDP, you could optionally configure it to provide a fallback LDP path in case the RSVP LSP becomes nonoperational:

```
[edit protocols]
ldp {
  interface so-1/1/0.0;
  interface at-2/0/0.0;
}
```


Enable RSVP and MPLS on the Provider Router

On the provider router, P2, you must configure RSVP and MPLS because this router exists on the MPLS LSP path between the provider Routers P1 and P3:

```
[edit]
protocols {
  rsvp {
    interface so-1/1/0.0;
    interface at-2/0/0.0;
  }
  mpls {
    interface so-1/1/0.0;
    interface at-2/0/0.0;
  }
}
```

Configure the MPLS LSP Tunnel between the Provider Routers

In this configuration example, LDP is tunneled over an RSVP LSP. Therefore, in addition to configuring RSVP, you must enable traffic engineering support in an IGP and you must create an MPLS LSP to tunnel the LDP traffic.

On Router P1, enable RSVP and configure one end of the MPLS LSP tunnel. In this example, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and to Router PE1. In the to statement, you specify the loopback address of Router P3.

```
[edit]
protocols {
  rsvp {
    interface so-1/0/1.0;
  }
  mpls {
    label-switched-path P1-to-P3 {
      to 10.255.100.1;
      ldp-tunneling;
    }
    interface so-1/0/0.0;
    interface so-1/0/1.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-1/0/0.0;
      interface so-1/0/1.0;
    }
  }
}
```

On Router P3, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and to Router PE2. In the to statement, you specify the loopback address of Router P1.

```
[edit]
protocols {
  rsvp {
    interface at-2/0/1.0;
  }
  mpls {
    label-switched-path P3-to-P1 {
      to 10.255.2.2;
      ldp-tunneling;
    }
    interface at-2/0/1.0;
    interface so-0/0/0.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface at-2/0/1.0;
      interface so-0/0/0.0;
    }
  }
}
```

Configure IBGP on the PE Routers

On the PE routers, configure an IBGP session with the following properties:

- **VPN family**—To indicate that the IBGP session is for the VPN, include the family inet-vpn statement.
- **Loopback address**—Include the local-address statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. Note that you must also configure the lo0 interface at the [edit interfaces] hierarchy level. The example does not include this part of the router's configuration.
- **Neighbor address**—Include the neighbor statement, specifying the IP address of the neighboring PE router, which is its loopback (lo0) address.

On Router PE1, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE1-to-PE2 {
      type internal;
      local-address 10.255.1.1;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.200.2;
    }
  }
}
```

On Router PE2, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE2-to-PE1 {
      type internal;
      local-address 10.255.200.2;
      family inet-vpn {
        unicast:
      }
      neighbor 10.255.1.1;
    }
  }
}
```

Configure Routing Instances for VPNs on the PE Routers

Both PE routers service VPN-A, so you must configure one routing instance on each router for the VPN in which you define the following:

- Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of vrf, which creates the VRF table on the PE router.
- Interfaces connected to the CE routers.
- VRF import and export policies, which must be the same on each PE router that services the same VPN. Unless the import policy contains only a then reject statement, it must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails.



Note

In this example, a private AS number is used for the route distinguisher. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

- Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—either BGP, OSPF, or RIP—or you can configure static routing.

On Router PE1, configure the following routing instance for VPN-A. In this example, Router PE1 uses RIP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-A {
    instance-type vrf;
    interface ge-1/0/0.0;
    route-distinguisher 65535:0;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    protocols {
      rip {
        group PE1-to-CE1 {
          neighbor ge-1/0/0.0;
        }
      }
    }
  }
}
```

On Router PE2, configure the following routing instance for VPN-A. In this example, Router PE2 uses OSPF to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-A {
    instance-type vrf;
    interface so-1/2/0.0;
    route-distinguisher 65535:1;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    protocols {
      ospf {
        area 0.0.0.0 {
          interface so-1/2/0.0;
        }
      }
    }
  }
}
```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which they use to forward packets within a VPN. For VPN-A, the VRF table is VPN-A.inet.0.

In the VPN policy, you also configure VPN target communities.



Note

In this example, a private AS number is used for the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

On Router PE1, configure the following VPN import and export policies.



The policy qualifiers shown in this example are only those needed for the VPN to function. You can configure additional qualifiers, as needed, to any policies that you configure.

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol rip;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:00;
}
```

On Router PE2, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

```

policy-statement VPN-A-export {
  term a {
    from protocol ospf;
    then {
      community add VPN-A;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community VPN-A members target:65535:00;

```

To apply the VPN policies on the routers, include the vrf-export and vrf-import statements when you configure the routing instance on the PE routers. The VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

LDP-over-MPLS VPN Configuration Summarized by Router

Router PE1

```

Routing Instance for VPN-A  routing-instance {
                             VPN-A {
                               instance-type vrf;
                               interface ge-1/0/0.0;
                               route-distinguisher 65535:0;
                               vrf-import VPN-A-import;
                               vrf-export VPN-A-export;
                             }
                           }

Instance Routing Protocol    protocols {
                             rip {
                               group PE1-to-CE1 {
                                 neighbor ge-1/0/0.0;
                               }
                             }
                           }

Interfaces                  interfaces {
                             so-1/0/0 {
                               unit 0 {
                                 family mpls;
                               }
                             }
                             ge-1/0/0 {
                               unit 0 {
                                 family mpls;
                               }
                             }
                           }

```

```

Master Protocol Instance protocols {

    Enable LDP    ldp {
                  interface so-1/0/0.0;
                  }

    Enable MPLS   mpls {
                  interface so-1/0/0.0;
                  interface ge-1/0/0.0;
                  }

    Configure IBGP bgp {
                  group PE1-to-PE2 {
                    type internal;
                    local-address 10.255.1.1;
                    family inet-vpn {
                      unicast;
                    }
                    neighbor 10.255.100.1;
                  }
                }

    Configure VPN Policy policy-options {
                      policy-statement VPN-A-import {
                        term a {
                          from {
                            protocol bgp;
                            community VPN-A;
                          }
                          then accept;
                        }
                        term b {
                          then reject;
                        }
                      }
                      policy-statement VPN-A-export {
                        term a {
                          from protocol rip;
                          then {
                            community add VPN-A;
                            accept;
                          }
                        }
                        term b {
                          then reject;
                        }
                      }
                      community VPN-A members target:65535:00;
                    }
  }

```

Router P1

```

Master Protocol Instance protocols {

    Enable RSVP   rsvp {
                  interface so-1/0/1.0;
                  }

```

Enable LDP

```
ldp {
  interface so-1/0/0.0;
  interface lo0.0;
}
```

Enable MPLS

```
mpls {
  label-switched-path P1-to-P3 {
    to 10.255.100.1;
    ldp-tunneling;
  }
  interface so-1/0/0.0;
  interface so-1/0/1.0;
}
```

Configure OSPF for Traffic Engineering Support

```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface so-1/0/0.0;
    interface so-1/0/1.0;
  }
}
```

Router P2

Master Protocol Instance

```
protocols {
```

Enable RSVP

```
  rsvp {
    interface so-1/1/0.0;
    interface at-2/0/0.0;
  }
```

Enable MPLS

```
  mpls {
    interface so-1/1/0.0;
    interface at-2/0/0.0;
  }
}
```

Router P3

Master Protocol Instance

```
protocols {
```

Enable RSVP

```
  rsvp {
    interface at-2/0/1.0;
  }
```

Enable LDP

```
  ldp {
    interface so-0/0/0.0;
    interface lo0.0;
  }
```

Enable MPLS

```
  mpls {
    label-switched-path P3-to-P1 {
      to 10.255.2.2;
      ldp-tunneling;
    }
    interface at-2/0/1.0;
    interface so-0/0/0.0;
  }
}
```


Configure OSPF for Traffic Engineering Support

```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface at-2/0/1.0;
    interface at-2/0/1.0;
  }
}
```

Router PE2**Routing Instance for VPN-A**

```
routing-instance {
  VPN-A {
    instance-type vrf;
    interface so-1/2/0.0;
    route-distinguisher 65535:1;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
  }
}
```

Instance Routing Protocol

```
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-1/2/0.0;
    }
  }
}
```

Interfaces

```
interfaces {
  so-0/0/0 {
    unit 0 {
      family mpls;
    }
  }
  so-1/2/0 {
    unit 0 {
      family mpls;
    }
  }
}
```

Master Protocol Instance

```
protocols {
```

Enable LDP

```
  ldp {
    interface so-0/0/0.0;
  }
```

Enable MPLS

```
  mpls {
    interface so-0/0/0.0;
    interface so-1/2/0.0;
  }
```

```

Configure IBGP    bgp {
                    group PE2-to-PE1 {
                        type internal;
                        local-address 10.255.200.2;
                        family inet-vpn {
                            unicast:
                        }
                        neighbor 10.255.1.1;
                    }
                }
    
```

```

Configure VPN Policy policy-options {
                    policy-statement VPN-A-import {
                        term a {
                            from {
                                protocol bgp;
                                community VPN-A;
                            }
                            then accept;
                        }
                        term b {
                            then reject;
                        }
                    }
                    policy-statement VPN-A-export {
                        term a {
                            from protocol ospf;
                            then {
                                community add VPN-A;
                                accept;
                            }
                        }
                        term b {
                            then reject;
                        }
                    }
                    community VPN-A members target:65535:01;
                }
    
```

Configure an Application-Based Layer 3 VPN Topology

This example illustrates an application-based mechanism for forwarding traffic into a Layer 3 VPN. Typically, one or more interfaces are associated with, or bound to, a VPN by including them in the configuration of the VPN routing instance. By binding the interface to the VPN, the VPN's VRF table is used to make forwarding decisions for any incoming traffic on that interface. Binding the interface also includes the interface local routes in the VRF, which provides next-hop resolution for VRF routes.

In this example, a firewall filter is used to define which incoming traffic on an interface is forwarded using the standard routing table, `inet.0`, and which incoming traffic is forwarded using the VRF table. You can expand this example such that incoming traffic on an interface can be redirected to one or more VPNs. For example, you can define a configuration to support a VPN that forwards traffic based on source address, that forwards HTTP traffic, or that forwards only streaming media.

For this configuration to work, the following must be true:

- The interfaces that use filter-based forwarding must not be bound to the VPN.
- Static routing must be used as the means of routing.
- You must define an interface routing table group that is shared among inet.0 and the VRFs to provide local routes to the VRF.

This example consists of two client hosts that are in two different VPNs and that want to send traffic both within the VPN and to the Internet. The paths are defined as follows:

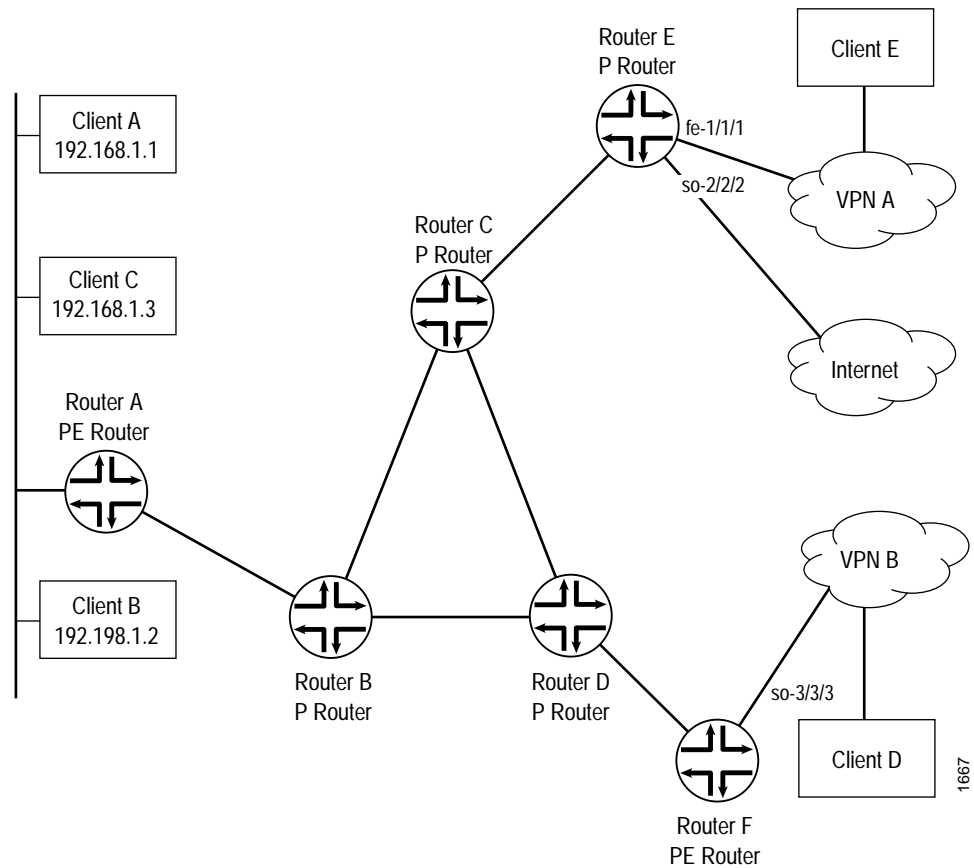
- Client A sends traffic to Client E over VPN A with a return path that also uses VPN A (using the VPN's VRF table).
- Client B sends traffic to Client D over VPN B with a return path that uses standard destination-based routing (using the inet.0 routing table).
- Clients B and C send traffic to the Internet using standard routing (using the inet.0 routing table), with a return path that also uses standard routing.

This example illustrates that there are a large variety of options in configuring an application-based Layer 3 VPN topology. This flexibility has application in many network implementations requiring specific traffic to be forwarded in a constrained routing environment.

This configuration example shows only the portions of the configuration for the filter-based forwarding, routing instances, and policy. It does not illustrate how to configure a Layer 3 VPN.

Figure 44 illustrates the configuration used in this example.

Figure 44: Application-Based Layer 3 VPN Example Configuration



Configuration on Router A

On Router A, you configure the interface to Clients A, B, and C. The configuration evaluates incoming traffic to determine whether it is to be forwarded using the VPN or using standard destination-based routing.

First, you apply an inbound filter and configure the interface to support MPLS.

```

interfaces {
  fe-1/1/0 {
    unit 0 {
      family inet {
        filter {
          input fbv-vrf;
        }
        address 192.168.1.1/24;
      }
      family mpls;
    }
  }
}

```

Because the interfaces that use filter-based forwarding must not be bound to a VPN, you must configure an alternate method to provide next-hop routes to the VRF table. You do this by defining an interface routing table group and sharing this group among all the routing tables. To provide a route back to the clients for normal inet.0 routing, you define a static route to include in inet.0 and redistribute the static route into BGP.

```
routing-options {
  interface-routes {
    rib-group inet if-rib;
  }
  static {
    route 192.168.1.0/24 next-hop fe-1/1/0.0
  }
  rib-groups {
    if-rib {
      import-rib [ inet.0 vpn-A.inet.0 vpn-B.inet.0 ];
    }
  }
}
```

You apply the following filter to incoming traffic on interface fe-1/1/0.0. The first term matches traffic from Client A and forwards it to the routing instance for VPN A. The second term matches traffic from Client B that is destined for Client D and forwards it to the routing instance for VPN B. The third term matches all other traffic, which is forwarded normally using destination-based forwarding according to the routes in inet.0.

```
filter fbf-vrf {
  term vpnA {
    from {
      source-address {
        192.168.1.1/32;
      }
    }
    then {
      routing-instance vpn-A;
    }
  }
  term vpnB {
    from {
      source-address {
        192.168.1.2/32;
      }
      destination-address {
        192.168.3.0/24;
      }
    }
    then routing-instance vpn-B;
  }
  term internet {
    then accept;
  }
}
```

You then configure the routing instances for VPN A and VPN B. Notice that these statements include all the required statements to define a Layer 3 VPN except for the interface statement.

```

routing-instances {
  vpn-A {
    instance-type vrf;
    route-distinguisher 172.21.10.63:100;
    vrf-import vpn-A-import;
    vrf-export vpn-A-export;
    routing-options {
      static {
        route 192.168.1.0/24 next-hop fe-1/1/0.0;
      }
    }
  }
  vpn-B {
    instance-type vrf;
    route-distinguisher 172.21.10.63:200;
    vrf-import vpn-B-import;
    vrf-export vpn-B-export;
    routing-options {
      static {
        route 192.168.1.0/24 next-hop fe-1/1/0.0;
      }
    }
  }
}

```

Configuration on Router E

On Router E, you configure a default route to reach the Internet. You should inject this route into the local IBGP mesh to provide an exit point from the network.

```

routing-options {
  static {
    route 0.0.0.0/0 next-hop so-2/2/2.0 discard
  }
}

```

For the interface to Client E, you configure it such that all incoming traffic on interface fe-1/1/1.0 that matches the VPN policy is forwarded over VPN A:

```

routing-instances {
  vpn-A {
    interface fe-1/1/1.0
    instance-type vrf;
    route-distinguisher 172.21.10.62:100;
    vrf-import vpn-A-import;
    vrf-export vpn-A-export;
    routing-options {
      static {
        route 192.168.2.0/24 next-hop fe-1/1/1.0;
      }
    }
  }
}

```

Configuration for Router F

Again, because the interfaces that use filter-based forwarding must not be bound to a VPN, you configure an alternate method to provide next-hop routes to the VRF table by defining an interface routing table group and sharing this group among all the routing tables. To provide a route back to the clients for normal inet.0 routing, you define a static route to include in inet.0 and redistribute the static route into BGP.

```
routing-options {
  interface-routes {
    rib-group inet if-rib;
  }
  rib-groups {
    if-rib {
      import-rib [ inet.0 vpn-B.inet.0];
    }
  }
}
```

To direct traffic from VPN B to Client D, you configure the routing instance for VPN B on Router F. All incoming traffic from Client D on interface so-3/3/3.0 is forwarded normally using the destination address based on the routes in inet.0.

```
routing-instances {
  vpn-B {
    instance-type vrf;
    route-distinguisher 172.21.10.64:200;
    vrf-import vpn-B-import;
    vrf-export vpn-B-export;
    routing-options {
      static {
        route 192.168.3.0/24 next-hop so-3/3/3.0;
      }
    }
  }
}
```

.....

Chapter 23

Summary of Layer 3 VPN Configuration Statements

The following sections explain the major routing-instances configuration statements that apply specifically to Layer 3 Virtual Private Networks (VPNs). The statements are organized alphabetically. Routing instances and the statements at the [edit routing-instances *routing-instance-name* routing-options] and [edit routing-instances *routing-instance-name* protocols] hierarchy levels are explained in the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

instance-type

Syntax	instance-type vrf;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Defines the type of routing instance.
Options	vrf—Virtual Routing and Forwarding instance. Required to create a VPN, creates a Virtual Routing and Forwarding (VRF) table (<i>instance-name</i> .inet.0), which contains the routes originating from and destined for a particular VPN. You must configure the interface, route-distinguisher, vrf-import, and vrf-export statements for this type of routing instance.
Usage Guidelines	See “Configure the Instance Type” on page 207.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

interface

Syntax	interface <i>interface-name</i> ;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Interface over which the VPN traffic travels between the provider edge (PE) router and customer edge (CE) router. You configure the interface on the PE router. If the instance type is vrf, the interface statement is required.
Usage Guidelines	See “Configure Interfaces for VPN Routing” on page 207.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

route-distinguisher

Syntax	route-distinguisher (<i>as-number:number</i> <i>ip-address:number</i>);
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Identifier attached to routes that distinguishes to which VPN it belongs. Each routing instance must have a unique route distinguisher associated with it. If the instance type is vrf, the route-distinguisher statement is required. The route distinguisher is a 6-byte value that you can specify in one of the following formats: <ul style="list-style-type: none"> ■ <i>as-number:number</i>, where <i>as-number</i> is your assigned AS number (a 2-byte value) and <i>number</i> is any 4-byte value. The AS number can be in the range of 1 through 65535. ■ <i>ip-address:number</i>, where <i>ip-address</i> is an IP address in your assigned prefix range (a 4-byte value) and <i>number</i> is any 2-byte value. The IP address can be any globally unique unicast address.
Usage Guidelines	See “Configure the Route Distinguisher” on page 208.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

vrf-export

Syntax	vrf-export [<i>policy-name</i>];
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	How routes are exported from the local PE router’s VRF table (<i>routing-instance-name.inet.0</i>) to the remote PE router. If the instance type is vrf, the vrf-export statement is required.
Options	You can configure multiple export policies on the PE.
Usage Guidelines	See “Configure Export Policy for the PE Router’s VRF Table” on page 211.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

vrf-import

Syntax	vrf-import [<i>policy name</i>];
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	How routes are imported into the local PE router’s VRF table (<i>routing-instance-name.inet.0</i>) from the remote PE router. If the instance type is vrf, the vrf-import statement is required.
Options	You can configure multiple import policies on the PE.
Usage Guidelines	See “Configure Import Policy for the PE Router’s VRF Table” on page 210.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

Chapter 24

Layer 2 VPN Overview

This chapter provides an overview of Layer 2 MPLS Virtual Private Networks (VPNs) as they are implemented in JUNOS software.

For information on Layer 3 VPNs and on VPNs in general, see “Layer 3 VPN Overview” on page 185.

This chapter discusses the following topics:

- Layer 2 VPN Overview on page 281
- Layer 2 VPN Standards on page 282

Layer 2 VPN Overview

A Layer 2 VPN is a set of sites that share common routing information and whose connectivity is controlled by a collection of policies. The sites that make up a VPN are connected over a service provider’s existing Internet backbone.

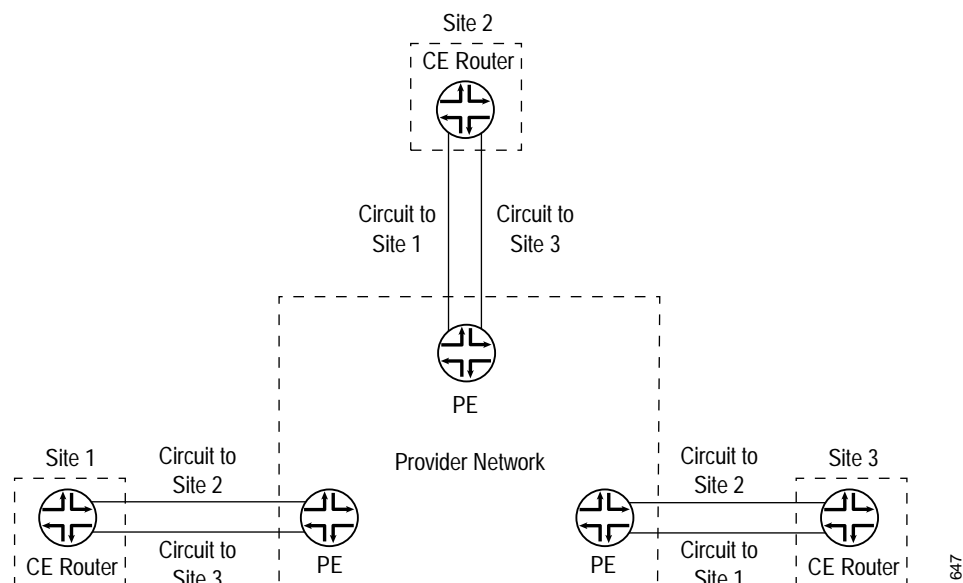
Running a Layer 2 VPN on a router is similar to running a VPN on an ATM or Frame Relay switch, except that the connection starts as ATM or Frame Relay, is carried by MPLS over the service provider’s network, and then is converted back to ATM or Frame Relay at the receiving site. The security and privacy of an MPLS Layer 2 VPN are equal to that of an ATM or Frame Relay VPN.

In contrast to a Layer 3 VPN, in which the routing occurs on the service provider’s routers, with Layer 2 VPNs, routing occurs on the customer’s routers, typically on the customer edge (CE) router. The CE router connected to a service provider on a Layer 2 VPN must select the appropriate circuit on which to send traffic. The provider edge (PE) router receiving the traffic sends it across the service provider’s network to the PE router connected to the receiving site. PE routers do not need to know the customer’s routes or routing topology; they need to know only in which tunnel to send the data.

In a Layer 2 VPN scenario, customers need to configure their own routers to carry all Layer 3 traffic. The service provider needs to know only how much traffic the Layer 2 VPN will need to carry. The service provider’s routers carry traffic between the customer’s sites using Layer 2 VPN interfaces. The VPN topology is determined by policies configured on the PE routers.

Customers need to know only which VPN interfaces connect to which of their own sites. Figure 45 illustrates a Layer 2 VPN in which each site has a VPN interface linked to each of the other customer sites.

Figure 45: Layer 2 VPN Connecting CE Routers



The benefits of implementing a Layer 2 MPLS VPN include:

- Service Providers do not have to invest in separate Layer 2 equipment to provide Layer 2 VPN service. A Layer 2 MPLS VPN allows providers to offer Layer 2 VPN service over an existing IP and MPLS backbone.
- You can configure the PE router to run any Layer 3 protocol in addition to the Layer 2 protocols.
- Customers who prefer to maintain control over most of the administration of their own networks might want Layer 2 VPN connections with their service provider instead of a Layer 3 VPN.

Layer 2 VPN Standards

Layer 2 VPNs are defined in the following documents:

- Internet draft draft-kompella-mpls-l2vpn-02.txt, *MPLS-based Layer 2 VPNs*.

To access Internet RFCs and drafts, go to the IETF Web site at <http://www.ietf.org>.

Chapter 25

Layer 2 VPN Configuration Guidelines

To configure Layer 2 Virtual Private Network (VPN) functionality, you must enable Layer 2 VPN support on the provider edge (PE) router. You must also configure PE routers to distribute routing information to the other PE routers in the VPN and configure the circuits between the PE routers and the customer edge (CE) routers.

You configure each Layer 2 VPN is configured under a routing instance of type l2vpn. An l2vpn routing instance can transparently carry Layer 3 traffic across the service provider's network. As with other routing instances, all logical interfaces belonging to a Layer 2 VPN routing instance are listed under that instance.

The configuration of the CE routers is not relevant to the service provider. The CE routers need to provision only appropriate Layer 2 circuits (with appropriate circuit identifiers, such as DLCI, VPI, or VCI) to send traffic to the PE router.

To configure Layer 2 VPNs, you include statements at the [edit routing-instances] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    instance-type l2vpn;
    interfaces {
      interface-name {
        encapsulation ccc-encapsulation-type;
        unit unit-number {
          encapsulation ccc-encapsulation-type;
        }
      }
    }
  }
}
route-distinguisher ( as-number:id | ip-address:id );
vrf-import [ policy-name ];
vrf-export [ policy-name ];
protocols {
  l2vpn {
    encapsulation-type <type>
    traceoptions {
      file filename <replace> <size size> <files number> <nostamp>;
      flag flag <flag-modifier> <disable>;
    }
  }
}
```

```

    site site-name {
        site-identifier identifier;
        interface interface-name {
            site-offset offset;
        }
    }
}

```

For Layer 2 VPNs, only some of the statements in the [edit routing-instances] hierarchy are valid. For the full hierarchy, see *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

In addition to these statements, you must configure MPLS Label Switched Paths (LSPs) between the PE routers, configure IBGP sessions between the PE routers, and configure an IGP on the PE and provider routers.

By default, Layer 2 VPNs are disabled.

This chapter describes the following tasks for configuring Layer 2 VPNs:

- Configure MPLS LSPs between the PE Routers on page 284
- Configure an IGP on PE and Provider Routers on page 288
- Configure an IBGP Session between PE Routers on page 288
- Configure Routing Instances for Layer 2 VPNs on the PE Routers on page 289
- Configure the Connections to the Local Site on page 291

Configure MPLS LSPs between the PE Routers

For Layer 2 VPNs to function, you must configure MPLS LSPs between the PE routers. You can do one of the following:

- Configure MPLS LSPs using LDP on page 285
- Configure MPLS LSPs Using RSVP on page 286

Configure MPLS LSPs using LDP

To use LDP to configure the MPLS LSPs, perform the following steps on the PE and provider routers:

1. Configure LDP on the interfaces in the core of the service provider's network by including the `ldp` statement at the `[edit protocols]` hierarchy level. You need to configure LDP only on the interfaces between PE routers or between PE and provider routers. You can think of these as the “core-facing” interfaces.

```
[edit]
protocols {
  ldp {
    interface interface-name;
  }
}
```

2. Configure the MPLS address family on the interfaces on which you enable LDP (that is, on the interfaces you configured in Step 1):

```
[edit]
interfaces {
  interface-name {
    unit logical-unit-number {
      family mpls;
    }
  }
}
```

Specify the interface name in the format *type-fpc/pic/port*.

3. Configure OSPF or IS-IS on each PE and provider router. You configure these protocols at the master instance of the routing protocol, not within the routing instance used for the VPN.

To configure OSPF, include the `ospf` statement at the `[edit protocols]` hierarchy level. At a minimum, you must configure a backbone area on at least one of the router's interfaces.

```
[edit]
protocols {
  ospf {
    area 0.0.0.0 {
      interface interface-name;
    }
  }
}
```

To configure IS-IS, include the `isis` statement at the `[edit protocols]` hierarchy level and configure the loopback interface and ISO family at the `[edit interfaces]` hierarchy level. At a minimum, you must enable IS-IS on the router, configure a network entity title (NET) on one of the router's interfaces (preferably the loopback interface, `lo0`), and configure the ISO family on all interfaces on which you want IS-IS to run. When you enable IS-IS, Level 1 and Level 2 are enabled by default. The following is the minimum IS-IS configuration. In the address statement, *address* is the NET.

```
[edit]
interfaces {
  lo0 {
    unit logical-unit-number {
      family iso {
        address address;
      }
    }
  }
  type-fpc/pic/port {
    unit logical-unit-number {
      family iso;
    }
  }
}
protocols {
  isis {
    interface all;
  }
}
```

For information about how to configure LDP, see “Configure LDP” on page 145. For more information about configuring OSPF and IS-IS, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Configure MPLS LSPs Using RSVP

To configure the MPLS LSPs using RSVP, perform the following steps:

1. On each PE router, configure traffic engineering. To do this, you must configure an IGP that supports traffic engineering (either IS-IS or OSPF) and enable traffic engineering support for that protocol.

To enable OSPF traffic engineering support, include the `traffic-engineering` statement at the `[edit protocols ospf]` hierarchy level:

```
[edit protocols ospf]
traffic-engineering;
```

For IS-IS, traffic engineering support is enabled by default.

2. On each PE and provider router, enable RSVP on the router interfaces that participate in the label-switched path (LSP). On the PE router, these are the interfaces that are the ingress and egress points to the LSP. On the provider router, these are the interfaces that connect the LSP between the PE routers.

To configure RSVP on the PE and provider routers, include the interface statement at the [edit rsvp] hierarchy level. Include one interface statement for each interface on which you are enabling RSVP.

```
[edit]
rsvp {
  interface interface-name;
}
```

3. On each PE router, configure an MPLS LSP to the PE router that is the LSP's egress point. To do this, include the label-switched-path and interface statements at the [edit mpls] hierarchy level.

```
[edit]
mpls {
  label-switched-path path-name {
    to ip-address;
  }
  interface interface-name;
}
```

In the to statement, specify the address of the LSP's egress point, which is an address on the remote PE router.

In the interface statement, specify the name of the interface (both the physical and logical portions). Include one interface statement for the interface associated with the LSP.

When you configure the same interface at the [edit interfaces] hierarchy level, you must also configure family mpls and family inet when configuring the logical interface:

```
[edit interfaces]
interface-name {
  unit logical-unit-number {
    family inet;
    family mpls;
  }
}
```

4. On all provider routers that participate in the LSP, enable MPLS by including the interface statement at the [edit mpls] hierarchy level. Include one interface statement for each connection to the LSP.

```
[edit]
mpls {
  interface interface-name;
  interface interface-name;
}
```

5. Enable MPLS on the interface between the PE and CE routers by including the interface statement at the [edit mpls] hierarchy level. Doing this allows the PE router to assign an MPLS label to traffic entering the LSP or to remove the label from traffic exiting the LSP.

```
[edit]
mpls {
  interface interface-name;
}
```

For information about configuring RSVP, see “RSVP Configuration Guidelines” on page 121. For information about configuring MPLS, see “Configure MPLS Signaled LSPs” on page 43, “Configure Static LSPs” on page 73, and “Configure Explicit-Path LSPs” on page 79.

Configure an IGP on PE and Provider Routers

To allow the PE and provider routers to exchange routing information, you must either configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the routing instance used for the Layer 2 VPN (that is, not at the [edit routing-instances] hierarchy level).

When you configure the PE router, do not configure any summarization of the PE router’s loopback addresses at the area boundary. Each PE router’s loopback address should appear as a separate route.

For information about configuring routing protocols and static routes, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Configure an IBGP Session between PE Routers

You must configure an IBGP session between PE routers to allow these routers to exchange information about Layer 2 VPNs, particularly information about sites connected to Layer 2 VPNs. The PE routers rely on this information to determine which labels to use for traffic destined for remote sites. To enable an IBGP session between the PE routers, include the family l2vpn statement when configuring IBGP in the master routing instance:

```
[edit protocols bgp]
bgp {
  group group-name {
    type internal;
    local-address ip-address;
    family l2vpn {
      unicast;
    }
    neighbor ip-address;
  }
}
```

The IP address in the local-address statement is the same as the address configured in the to statement at the [edit protocols mpls label-switched-path *lsp-path-name*] hierarchy level on the remote PE router. The IBGP session uses this address as the source in the peering session.

The IP address in the neighbor statement is the loopback address of the neighboring PE router. If you are using RSVP signaling, this IP address is the same address you specify in the to statement at the [edit mpls label-switched-path] hierarchy level when you configure the MPLS LSP.

Configure Routing Instances for Layer 2 VPNs on the PE Routers

To configure routing instances for Layer 2 VPNs, include the `routing-instances` statement at the `[edit]` hierarchy level. You configure Layer 2 VPN routing instances only on the PE routers.

```
[edit]
routing-instances {
  routing-instance-name {
    instance-type l2vpn;
    interface interface-name;
    route-distinguisher ( as-number:id | ip-address:id );
    vrf-export [ policy-name ];
    vrf-import [ policy-name ];
  }
}
```



Note

For the Layer 2 VPN to function, you must include the `instance-type`, `interface`, `route-distinguisher`, `vrf-export`, and `vrf import` statements in the routing instance configuration on the PE router.

The following sections describe how to configure Layer 2 VPN routing instances:

- Configure the Instance Type on page 289
- Configure Interfaces for Layer 2 VPN Routing on page 289
- Configure CCC Encapsulation on Interfaces on page 290
- Configure the Route Distinguisher on page 291
- Configure Policy for the PE Router's VRF Table on page 291

Configure the Instance Type

To enable Layer 2 VPN routing on a PE router, include the `instance-type` statement at the `[edit routing-instances routing-instance-name]` hierarchy level, specifying the instance type as `l2vpn`:

```
[edit routing-instances routing-instance-name]
instance-type l2vpn;
```

Configure Interfaces for Layer 2 VPN Routing

On each PE router, you must configure the interfaces over which the Layer 2 VPN traffic travels between PE and CE routers. To do this, include the `interface` statement at the `[edit routing-instances routing-instance-name]` hierarchy level:

```
[edit routing-instances routing-instance-name]
interface interface-name;
```

You should specify both the physical and logical portions of the interface name, in the following format:

physical.logical

For example, in atm-1/2/1.2, atm-1/2/1 is the physical portion of the interface name and 2 is the logical portion. If you do not specify the logical portion of the interface name, 0 is set by default.

A logical interface can be associated with only one routing instance.



Note

If you enable a routing protocol on all instances by specifying interfaces all when configuring the master instance of the protocol at the [edit protocols] hierarchy level and if you configure a specific interface for Layer 2 VPN routing at the [edit routing-instances *routing-instance-name*] hierarchy level, the latter interface statement takes precedence and the interface is used exclusively for the Layer 2 VPN.

If you explicitly configure the same interface name at both the [edit protocols] and [edit routing-instances *routing-instance-name*] hierarchy levels, when you try to commit the configuration, it will fail.

Configure CCC Encapsulation on Interfaces

You need to specify a CCC encapsulation type for each PE router to CE router interface running a Layer 2 VPN. This encapsulation type should match the encapsulation type configured under the routing instance. See “Configure the Encapsulation Type” on page 292 for information about how to configure the encapsulation type under the routing instance.

To configure the ccc encapsulation type, include the following statements:

```
[edit]
interfaces {
  interface name {
    encapsulation ccc-encapsulation-type;
  }
  unit unit number {
    encapsulation ccc-encapsulation-type;
  }
}
```

You configure the encapsulation type at the [edit interfaces] hierarchy level differently than you do at the [edit routing-instance] hierarchy level. For example, you specify the encapsulation as frame-relay at the [edit routing-instances] hierarchy level and you specify the CCC encapsulation as frame-relay-ccc at the [edit interfaces] hierarchy level.

You can run both standard Frame Relay and CCC Frame Relay on the same device. If you specify Frame Relay encapsulation (frame-relay-ccc) for the interface, you should also configure the encapsulation at the [edit interfaces *interface name* unit *unit-number*] as frame-relay-ccc. Otherwise the logical interface unit defaults to standard Frame Relay.

Configure the Route Distinguisher

Each routing instance that you configure on a PE router must have a unique route distinguisher associated with it. With Layer 2 VPNs, the route distinguisher helps BGP distinguish overlapping NLRIs from different VPNs.

We recommend that you use unique route distinguishers for each routing instance that you configure. Although you could use the same route distinguisher on all PE routers in the same Layer 2 VPN, if you use a unique route distinguisher, you can determine the PE router from which a route originated.

To configure a route distinguisher on a PE router, include the `route-distinguisher` statement at the `[edit routing-instances routing-instance-name]` hierarchy level:

```
[edit routing-instances routing-instance-name]  
route-distinguisher ( as-number:number | ip-address:number );
```

The route distinguisher is a 6-byte value that you can specify in one of the following formats:

- *as-number:number*, where *as-number* is an AS number (a 2-byte value) and *number* is any 4-byte value. The AS number can be in the range 1 through 65535. We recommend that you use an IANA assigned, nonprivate AS number, preferably the ISP's own or the customer's own AS number.
- *ip-address:number*, where *ip-address* is an IP address (a 4-byte value) and *number* is any 2-byte value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the `router-id` statement, which is a nonprivate address in your assigned prefix range.

Configure Policy for the PE Router's VRF Table

For information about configuring the VRF table, see "Configure Policy for the PE Router's VRF Table" on page 291.

Configure the Connections to the Local Site

For each local site, the PE router advertises a set of VPN labels to the other PE routers servicing the Layer 2 VPN. The VPN labels comprise a single block of contiguous labels. However, to allow for reprovisioning, more than one such block can be advertised. Each label block consists of a label base, a range (the size of the block), and a site offset that identifies the sequence of remote sites that connect to the local site using this label block (the site offset is the first site identifier in the sequence). The encapsulation type is also advertised along with the label block.

The following sections describe how to configure the connections to the local site on the PE router:

- Configure the Local Site on page 292
- Configure the Encapsulation Type on page 292
- Examine Layer 2 VPN Traffic Using Trace Options on page 293

Configure the Local Site

On each PE router, you must configure each site that has a Layer 2 circuit to the PE router. To do this, include the site statement at the [edit routing-instances *routing-instance-name* protocols l2vpn] hierarchy level:

```
[edit routing-instances routing-instance-name protocols l2vpn]
site site-name {
  site-identifier identifier;
  interface interface-name {
    site-offset offset;
  }
}
```

For each site, you configure the following:

In the site statement, specify the name of the site.

The identifier in the site-identifier statement is an unsigned 16-bit number greater than zero that uniquely identifies the site.

In the interface statement, specify name for the interface and optionally specify a site offset for remote site connections.

The site offset allows for a sparse Layer 2 VPN topology. When you configure site offsets, each site does not have to connect to all other sites in the Layer 2 VPN, making it unnecessary to allocate circuits for all of the remote sites. Site offsets are particularly important if you configure a topology more complicated than full mesh.

If you omit the site offset, it is set to one higher than the site offset for the previous interface. For example, if the first interface in the list does not have a site offset, its offset is set to 1. The second interface in the list would have its offset set to 2, and the third would have its offset set to 3, and so on.

Configure the Encapsulation Type

All Layer 2 VPN sites must use the same Layer 2 protocol. Therefore, you must configure the Layer 2 VPN routing instance of each PE router with the same encapsulation type.

To configure the Layer 2 protocol accepted by the PE router, you specify the encapsulation type by including the encapsulation-type statement at the [edit routing-instances *routing-instance-name* protocols l2vpn] hierarchy level:

```
[edit routing-instances routing-instance-name protocols l2vpn]
encapsulation-type type
```

The following Layer 2 encapsulation types are supported:

- atm-aal5—ATM AAL/5
- atm-cell—ATM cell
- cisco-hdlc—Cisco Systems compatible HDLC

- ethernet-vlan—Ethernet VLAN
- frame-relay—Frame Relay
- ppp—PPP

Examine Layer 2 VPN Traffic Using Trace Options

To trace Layer 2 VPN protocol traffic, include the traceoptions statement at the [edit routing-instances *routing-instance-name* protocols l2vpn] hierarchy level:

```
[edit routing-instances routing-instance-name protocols l2vpn]
traceoptions {
  file filename <replace> <size size> <files number> <nostamp>;
  flag flag <flag-modifier> <disable>;
}
```

The following trace flags display the operations associated with Layer 2 VPNs:

- all—All Layer 2 VPN tracing options.
- connections—Layer 2 VPN connections (events and state changes).
- error—Error conditions.
- nlri—Layer 2 VPN advertisements received or sent using BGP.
- route—Routing information.
- topology—Layer 2 VPN topology changes due to reconfiguration or due to advertisements received from other PE routers using BGP.

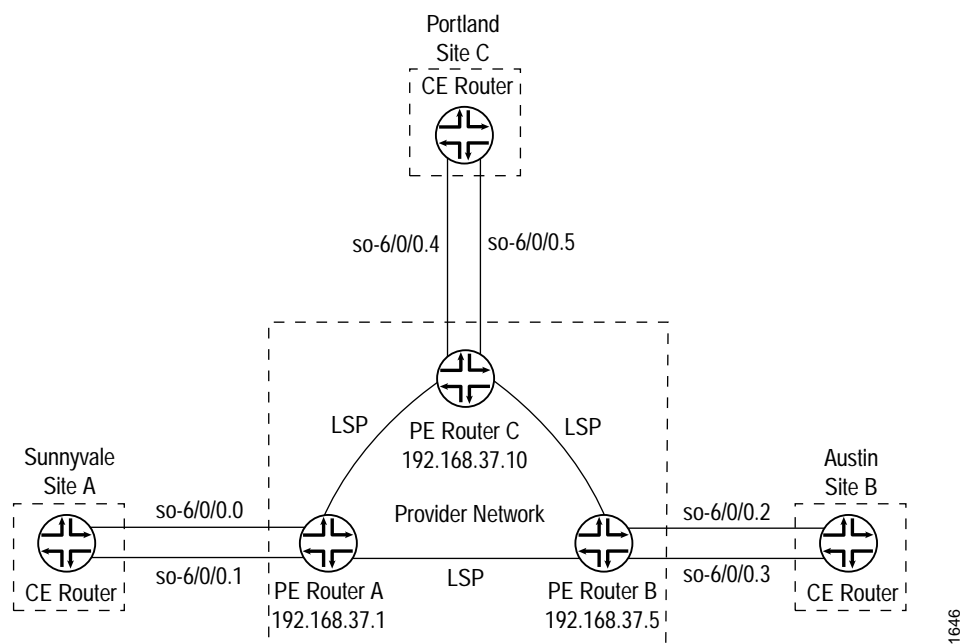
.....

Chapter 26

Layer 2 VPN Configuration Example

This chapter illustrates how to configure a simple Layer 2 VPN. In the example in this chapter, you configure a Layer 2 VPN spanning three sites: Sunnyvale, Austin, and Portland. Each site has a PE router connected to it. The CE routers at each of the sites use Frame Relay for Layer 2 traffic to the PE routers. Figure 46 illustrates the topology of this Layer 2 VPN.

Figure 46: Example of a Simple Layer 2 VPN Topology



The following sections explain how to configure Layer 2 VPN functionality on the PE routers connected to each of the sites.

- Enable an IGP on the PE routers on page 296
- Configure MPLS LSP Tunnels between the PE Routers on page 296
- Configure IBGP on the PE Routers and Provider Routers on page 298
- Configure Routing Instances for Layer 2 VPNs on the PE Routers on page 299

- Configure VPN Policy on the PE Routers on page 302
- Layer 2 VPN Configuration Summarized by Router on page 304

Enable an IGP on the PE routers

To allow the PE routers to exchange routing information among themselves, you must configure an IGP on these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the Layer 2 VPN routing instance (that is, not at the [edit routing-instances] hierarchy level). Turn on traffic engineering on the IGP.

You configure the IGP in the standard way. This example does not include this portion of the configuration.

Configure MPLS LSP Tunnels between the PE Routers

In this configuration example, RSVP is used for MPLS signaling. Therefore, in addition to configuring RSVP, you must create an MPLS LSP to tunnel the VPN traffic.

On Router A, enable RSVP and configure one end of the MPLS LSP tunnel to Router B. When configuring the MPLS LSP, include all interfaces using the interface all statement.

```
[edit]
protocols {
  rsvp {
    interface all;
  }
  mpls {
    label-switched-path RouterA-to-RouterB {
      to 192.168.37.5;
      primary Path-to-RouterB {
        cspf
      }
    }
    label-switched-path RouterA-to-RouterC {
      to 192.168.37.10;
      primary Path-to-RouterC {
        cspf
      }
    }
    interface all;
  }
}
```

On Router B, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, configure all of the interfaces using the interface all statement.

```
[edit]
protocols {
  rsvp {
    interface all;
  }
}
```

```

mpls {
  label-switched-path RouterB-to-RouterA {
    to 192.168.37.1;
    primary Path-to-RouterA {
      cspf;
    }
  }
  label-switched-path RouterB-to-RouterC {
    to 192.168.37.10;
    primary Path-to-RouterC {
      cspf;
    }
  }
  interface all;
}

```

On Router C, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, configure all of the interfaces using the interface all statement.

```

[edit]
protocols {
  rsvp {
    interface all;
  }
  mpls {
    label-switched-path RouterC-to-RouterA {
      to 192.168.37.1;
      primary Path-to-RouterA {
        cspf;
      }
    }
    label-switched-path RouterC-to-RouterB {
      to 192.168.37.5;
      primary Path-to-RouterB {
        cspf;
      }
    }
    interface all;
  }
}

```

Configure IBGP on the PE Routers and Provider Routers

On the PE routers, configure an IBGP session with the following properties:

- **Layer 2 VPN**—To indicate that the IBGP session is for a Layer 2 VPN, include the family l2vpn statement.
- **Local address**—The IP address in the local-address statement is the same as the address configured in the to statement at the [edit protocols mpls label-switched-path *lsp-path-name*] hierarchy level. The IBGP session for Layer 2 VPNs runs through this address.
- **Neighbor address**—Include the neighbor statement, specifying the IP address of the neighboring PE router.

On Router A, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    import match-all;
    export match-all;
    group pe-pe {
      type internal;
      neighbor 192.168.37.5 {
        local-address 192.168.37.1;
        family l2vpn {
          unicast;
        }
      }
      neighbor 192.168.37.10 {
        local-address 192.168.37.1;
        family l2vpn {
          unicast;
        }
      }
    }
  }
}
```

On Router B, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    local-address 192.168.37.5;
    import match-all;
    export match-all;
    group pe-pe {
      type internal;
      neighbor 192.168.37.1 {
        local-address 192.168.37.5;
        family l2vpn {
          unicast;
        }
      }
    }
  }
}
```

```

neighbor 192.168.37.10 {
  local-address 192.168.37.5;
  family l2vpn {
    unicast;
  }
}
}
}

```

On Router C, configure IBGP as follows:

```

[edit]
protocols{
  bgp {
    local-address 192.168.37.10;
    import match-all;
    export match-all;
    group pe-pe {
      type internal;
      neighbor 192.168.37.1 {
        local-address 192.168.37.10;
        family l2vpn {
          unicast;
        }
      }
      neighbor 192.168.37.5 {
        local-address 192.168.37.10;
        family l2vpn {
          unicast;
        }
      }
    }
  }
}
}

```

Configure Routing Instances for Layer 2 VPNs on the PE Routers

The three PE routers service the Layer 2 VPN, so you need to configure a routing instance on each router. For the VPN, you must define the following in the routing instance:

- Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of l2vpn, which configures the router to run a Layer 2 VPN.
- Interfaces connected to the CE routers.
- VRF import and export policies, which must be the same on each PE router that services the same VPN and are used to control the network topology. Unless the import policy contains only a then reject statement, it must include a reference to a community. Otherwise, when you attempt to commit the configuration, the commit fails.

On Router A, configure the following routing instances for the Layer 2 VPN:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.0 {
      encapsulation frame-relay-ccc;
      unit 0 {
        encapsulation frame-relay-ccc;
      }
    }
    interface so-6/0/0.1 {
      encapsulation frame-relay-ccc;
    }
    unit 1 {
      encapsulation frame-relay-ccc;
    }
  }
  route-distinguisher 100:1;
  vrf-import vpn-SPA-import;
  vrf-export vpn-SPA-export;
  protocols {
    l2vpn {
      encapsulation-type frame-relay;
      site Sunnyvale {
        site-identifier 1;
        interface so-6/0/0.0;
        interface so-6/0/0.1;
      }
    }
  }
}
```

On Router B, configure the following routing instance:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.2 {
      encapsulation frame-relay-ccc;
      unit 2 {
        encapsulation frame-relay-ccc;
      }
    }
    interface so-6/0/0.3 {
      encapsulation frame-relay-ccc;
    }
    unit 3 {
      encapsulation frame-relay-ccc;
    }
  }
  route-distinguisher 100:1;
  vrf-import vpn-SPA-import;
  vrf-export vpn-SPA-export;
```

```

protocols {
  l2vpn {
    encapsulation-type frame-relay;
    site Austin {
      site-identifier 2;
      interface so-6/0/0.2;
      interface so-6/0/0.3;
    }
  }
}

```

On Router C, configure the following routing instance for the Layer 2 VPN:

```

[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.4 {
      encapsulation frame-relay-ccc;
      unit 4 {
        encapsulation frame-relay-ccc;
      }
    }
    interface so-6/0/0.5 {
      encapsulation frame-relay-ccc;
    }
    unit 5 {
      encapsulation frame-relay-ccc;
    }
  }
  route-distinguisher 100:1;
  vrf-import vpn-SPA-import;
  vrf-export vpn-SPA-export;
  protocols {
    l2vpn {
      encapsulation-type frame-relay;
      site Portland {
        site-identifier 3;
        interface so-6/0/0.4;
        interface so-6/0/0.5;
      }
    }
  }
}

```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which they use to forward packets within the VPN.

On Router A, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-export {
    term a {
      then {
        community add SPA-com;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  community SPA-com members target:69:100;
}
```

On Router B, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```



```

policy-statement vpn-SPA-export {
  term a {
    then {
      community add SPA-com;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community SPA-com members target:69:100;
}

```

On Router C, configure the following VPN import and export policies:

```

[edit]
policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-SPA-export {
    term a {
      then {
        community add SPA-com;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community SPA-com members target:69:100;
}

```

To apply the VPN policies on the routers, include the `vrf-export` and `vrf-import` statements when you configure the routing instance. The VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

To apply the VPN policies on Router A, include the following statements:

```

[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}

```

To apply the VPN policies on Router B, include the following statements:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}
```

To apply the VPN policies on Router C, include the following statements:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}
```

Layer 2 VPN Configuration Summarized by Router

Summary for Router A (PE Router for Sunnyvale)

```
Routing Instance for VPN routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.0 {
      encapsulation frame-relay-ccc;
      unit 0 {
        encapsulation frame-relay-ccc;
      }
    }
    interface so-6/0/0.1 {
      encapsulation frame-relay-ccc;
    }
    unit 1 {
      encapsulation frame-relay-ccc;
    }
  }
  route-distinguisher 100:1;
  vrf-import vpn-SPA-import;
  vrf-export vpn-SPA-export;
```

```
Configure Layer 2 VPN protocols {
  l2vpn {
    encapsulation-type frame-relay;
    site Sunnyvale {
      site-identifier 1;
      interface so-6/0/0.0;
      interface so-6/0/0.1;
    }
  }
}
```

```

Master Protocol Instance protocols {

    Enable RSVP      rsvp {
                        interface all;
                    }

    Configure MPLS LSPs mpls {
                        label-switched-path RouterA-to-RouterB {
                            to 192.168.37.5;
                            primary Path-to-RouterB {
                                cspf;
                            }
                        }
                        label-switched-path RouterA-to-RouterC {
                            to 192.168.37.10;
                            primary Path-to-RouterC {
                                cspf;
                            }
                        }
                        interface all;
                    }

    Configure IBGP    bgp {
                        import match-all;
                        export match-all;
                        group pe-pe {
                            type internal;
                            neighbor 192.168.37.5 {
                                local-address 192.168.37.1;
                                family l2vpn {
                                    unicast;
                                }
                            }
                            neighbor 192.168.37.10 {
                                local-address 192.168.37.1;
                                family l2vpn {
                                    unicast;
                                }
                            }
                        }
                    }

    Configure VPN Policy policy-options {
                        policy-statement match-all {
                            term acceptable {
                                then accept;
                            }
                        }
                        policy-statement vpn-SPA-export {
                            term a {
                                then {
                                    community add SPA-com;
                                    accept;
                                }
                            }
                            term b {
                                then reject;
                            }
                        }
                    }

```

```

policy-statement vpn-SPA-import {
  term a {
    from {
      protocol bgp;
      community SPA-com;
    }
    then accept;
  }
  term b {
    then reject;
  }
}
community SPA-com members target:69:100;
}

```

Summary for Router B (PE Router for Austin)

```

Routing Instance for VPN routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.2 {
      encapsulation frame-relay-ccc;
      unit 2 {
        encapsulation frame-relay-ccc;
      }
    }
    interface so-6/0/0.3 {
      encapsulation frame-relay-ccc;
    }
    unit 3 {
      encapsulation frame-relay-ccc;
    }
  }
  route-distinguisher 100:1;
  vrf-import vpn-SPA-import;
  vrf-export vpn-SPA-export;
}

```

```

Configure Layer 2 VPN protocols {
  l2vpn {
    encapsulation-type frame-relay;
    site Austin {
      site-identifier 2;
      interface so-6/0/0.2;
      interface so-6/0/0.3;
    }
  }
}

```

```

Master Protocol Instance protocols {

```

```

  Enable RSVP rsvp {
    interface all;
  }
}

```

Configure MPLS LSPs

```

mpls {
  label-switched-path RouterB-to-RouterA {
    to 192.168.37.1;
    primary Path-to-RouterA {
      cspf;
    }
  }
  label-switched-path RouterB-to-RouterC {
    to 192.168.37.10;
    primary Path-to-RouterC {
      cspf;
    }
  }
  interface all;
}

```

Configure IBGP

```

bgp {
  local-address 192.168.37.5;
  import match-all;
  export match-all;
  group pe-pe {
    type internal;
    neighbor 192.168.37.1 {
      local-address 192.168.37.5;
      family I2vpn {
        unicast;
      }
    }
    neighbor 192.168.37.10 {
      local-address 192.168.37.5;
      family I2vpn {
        unicast;
      }
    }
  }
}

```

Configure VPN Policy

```

policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}

```

```

policy-statement vpn-SPA-export {
  term a {
    then {
      community add SPA-com;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community SPA-com members target:69:100;
}

```

Summary for Router C (PE Router for Portland)

```

Routing Instance for VPN routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.3 {
      encapsulation frame-relay-ccc;
      unit 3 {
        encapsulation frame-relay-ccc;
      }
    }
    interface so-6/0/0.4 {
      encapsulation frame-relay-ccc;
    }
    unit 4 {
      encapsulation frame-relay-ccc;
    }
  }
  route-distinguisher 100:1;
  vrf-import vpn-SPA-import;
  vrf-export vpn-SPA-export;
}

```

```

Configure Layer 2 VPN protocols {
  l2vpn {
    encapsulation-type frame-relay;
    site Portland {
      site-identifier 3;
      interface so-6/0/0.3;
      interface so-6/0/0.4;
    }
  }
}

```

```

Master Protocol Instance protocols {

```

```

  Enable RSVP rsvp {
    interface all;
  }

```

```

  Configure MPLS LSPs mpls {
    label-switched-path RouterC-to-RouterA {
      to 192.168.37.1;
      primary Path-to-RouterA {
        cspf;
      }
    }
  }
}

```

```

label-switched-path RouterC-to-RouterB {
  to 192.168.37.5;
  primary Path-to-RouterB {
    cspf;
  }
}
interface all;
}

Configure IBGP
bgp {
  local-address 192.168.37.10;
  import match-all;
  export match-all;
  group pe-pe {
    type internal;
    neighbor 192.168.37.1 {
      local-address 192.168.37.10;
      family l2vpn {
        unicast;
      }
    }
    neighbor 192.168.37.5 {
      local-address 192.168.37.10;
      family l2vpn {
        unicast;
      }
    }
  }
}

Configure VPN Policy
policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-SPA-export {
    term a {
      then {
        community add SPA-com;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community SPA-com members target:69:100;
}

```

.....

Chapter 27

Summary of Layer 2 VPN Configuration Statements

The following sections explain the major routing-instances configuration statements that apply specifically to Layer 2 Virtual Private Networks (VPNs). The statements are organized alphabetically. Routing instances and the statements at the [edit routing-instances *routing-instance-name* protocols] hierarchy level are explained in the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

encapsulation-type

encapsulation-type (ccc)

Syntax	encapsulation-type <i>type</i>
Hierarchy Level	[edit interfaces <i>interface name</i> encapsulation]
Description	Layer 2 protocol used for traffic from the CE router. The encapsulation type configured here is the CCC encapsulation type. Note that an encapsulation type must also be configured at the [edit routing-instances <i>routing-instance-name</i> protocols l2vpn] hierarchy level.
Options	<i>type</i> —The following Layer 2 CCC encapsulation types are supported: <ul style="list-style-type: none">■ atm-aal5-ccc—ATM AAL/5.■ atm-cell-ccc—ATM cell.■ cisco-hdlc-ccc—Cisco Systems compatible HDLC.■ ethernet-vlan-ccc—Ethernet VLAN.■ frame-relay-ccc—Frame Relay.■ ppp-ccc—PPP.

You can run both standard Frame Relay and CCC Frame Relay on the same device. If you specify Frame Relay encapsulation (*frame-relay-ccc*) for the interface, you should also configure the encapsulation at the [edit interfaces *interface name* unit *unit-number*] hierarchy level as *frame-relay-ccc*. Otherwise, the logical interface unit defaults to standard Frame Relay.

Usage Guidelines See “Configure MPLS LSPs between the PE Routers” on page 284.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

See Also encapsulation-type (layer 2 vpn) on page 312

encapsulation-type (layer 2 vpn)

Syntax encapsulation-type <type>

Hierarchy Level [edit routing-instances *routing-instance-name* protocols l2vpn]

Description Layer 2 protocol used for traffic from the CE router.

Options <type>—The following Layer 2 encapsulation types are supported:

- atm-aal5—ATM AAL/5.
- atm-cell—ATM cell.
- cisco-hdlc—Cisco Systems compatible HDLC.
- ethernet-vlan—Ethernet VLAN.
- frame-relay—Frame Relay.
- ppp—PPP.

Usage Guidelines See “Configure the Encapsulation Type” on page 292.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

See also encapsulation-type (ccc) on page 311

instance-type

Syntax instance-type l2vpn;

Hierarchy Level [edit routing-instances *routing-instance-name*]

Description The type of routing instance.

Options l2vpn—Layer 2 VPN instance.

Usage Guidelines See “Configure the Instance Type” on page 289.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

interface

Syntax	interface <i>interface-name</i> ;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i> protocols l2vpn]
Description	Interface over which Layer 2 VPN traffic travels between the provider edge (PE) router and the customer edge (CE) router. You configure the interface on the PE router. If the instance type is l2vpn, the interface statement is required.
Options	<i>interface-name</i> —Name of the interface to configure.
Usage Guidelines	See “Configure Interfaces for Layer 2 VPN Routing” on page 289.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

route-distinguisher

Syntax	route-distinguisher (<i>ip-address:number</i> <i>as-number:number</i>);
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Identifier attached to routes that distinguishes to which VPN it belongs. Each routing instance must have a unique distinguisher associated with it. Each route distinguisher is a 6-byte value.
Options	<i>as-number:number</i> — <i>as-number</i> is your assigned AS number (a 2-byte value) and <i>number</i> is any 4-byte value. The AS number can be in the range of 1 through 65535. <i>ip-address:number</i> — <i>ip-address</i> is an IP address in your assigned prefix range (a 4-byte value) and <i>number</i> is any 2-byte value. The IP address can be any globally unique unicast address.
Usage Guidelines	See “Configure the Route Distinguisher” on page 291.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

site

Syntax site *site-name* {
 site-identifier *identifier*;
 interface *interface-name* {
 site-offset *offset*;
 }
 }

Hierarchy Level [edit routing-instances *routing-instance-name* protocols l2vpn]

Description Specifies the site name, site identifier, and interfaces connecting to the site and allows you to configure a site offset for remote sites.

Options interface *interface-name*—Name of the interface.

site-identifier *identifier*—Numerical identifier for the site used as a default reference for the site offset.

site-offset *offset*—(Optional) Control the remote interface to which the interface should connect. The order of the interfaces configured for the site determines the default value if you do not explicitly configure the site offset.

site *site-name*—Name of the site.

The remaining statements are explained separately.

Usage Guidelines See “Configure the Local Site” on page 292.

Required Privilege Level routing—To view this statement in the configuration.
 routing-control—To add this statement to the configuration.

traceoptions

Syntax traceoptions {
 file *filename* <replace> <size *size*> <files *number*> <nostamp>;
 flag *flag* <flag-modifier> <disable>;
 }

Hierarchy Level [edit routing-instances *routing-instance-name* protocols l2vpn]

Description Trace traffic flowing through a Layer 2 VPN.

Options disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks.

files *number*—(Optional) Maximum number of trace files. When a trace file named *trace-file* reaches its maximum size, it is renamed *trace-file.0*, then *trace-file.1*, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the *size* option.

Range: 2 to 1000

Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple *flag* statements.

- **all**—All Layer 2 VPN tracing options.
- **connections**—Layer 2 VPN connections (events and state changes).
- **error**—Error conditions.
- **nlri**—Layer 2 VPN advertisements received or sent using BGP.
- **route**—Routing information.
- **topology**—Layer 2 VPN topology changes due to reconfiguration or due to advertisements received from other PE routers using BGP.

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- **detail**—Provide detailed trace information.

no stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file.0*. When the *trace-file* again reaches its maximum size, *trace-file.0* is renamed *trace-file.1* and *trace-file* is renamed *trace-file.0*. This renaming scheme continues until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the *files* option.

Syntax: *xk* to specify KB, *xm* to specify MB, or *xg* to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

Usage Guidelines See “Examine Layer 2 VPN Traffic Using Trace Options” on page 293.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vrf-export

Syntax vrf-export [*policy-name*];

Hierarchy Level [edit routing-instances *routing-instance-name*]

Description How routes are exported from the local PE router's VRF table (*routing-instance-name.inet.0*) to the remote PE router. If the instance type is vrf, the vrf-export statement is required.

Options You can configure multiple export policies on the PE.

Usage Guidelines See "Configure Export Policy for the PE Router's VRF Table" on page 211.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vrf-import

Syntax vrf-import [*policy name*];

Hierarchy Level [edit routing-instances *routing-instance-name*]

Description How routes are imported into the local PE router's VRF table (*routing-instance-name.inet.0*) from the remote PE router. If the instance type is vrf, the vrf-import statement is required.

Options You can configure multiple import policies on the PE.

Usage Guidelines See "Configure Import Policy for the PE Router's VRF Table" on page 210.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Part 7

Appendix

■ Glossary on page 319

.....

Appendix A

Glossary

A

AAL	ATM adaptation layer. A series of protocols enabling various types of traffic, including voice, data, image, and video, to run over an ATM network.
active route	Route chosen from all routes in the routing table to reach a destination. Active routes are installed into the forwarding table.
add/drop multiplexer	See ADM.
Address Resolution Protocol	See ARP.
adjacency	Portion of the local routing information that pertains to the reachability of a single neighbor over a single circuit or interface.
ADM	Add/drop multiplexer. SONET functionality that allows lower-level signals to be dropped from a high-speed optical connection.
aggregation	Combination of groups of routes that have common addresses into a single entry in the routing table.
ANSI	American National Standards Institute. The United States' representative to the ISO.
APS	Automatic Protection Switching. Technology used by SONET ADMs to protect against circuit faults between the ADM and a router and to protect against failing routers.
area	<p>Routing subdomain that maintains detailed routing information about its own internal composition and that maintains routing information that allows it to reach other routing subdomains. In IS-IS, an area corresponds to a Level 1 subdomain.</p> <p>In IS-IS and OSPF, a set of contiguous networks and hosts within an autonomous system that have been administratively grouped together.</p>
area border router	Router that belongs to more than one area. Used in OSPF.
ARP	Address Resolution Protocol. Protocol for mapping IP addresses to MAC addresses.
AS	Autonomous system. Set of routers under a single technical administration. Each AS normally uses a single interior gateway protocol (IGP) and metrics to propagate routing information within the set of routers. Also called <i>routing domain</i> .
AS boundary router	In OSPF, routers that exchange routing information with routers in other ASs.

AS external link advertisements	OSPF link-state advertisement sent by AS boundary routers to describe external routes that they know. These link-state advertisements are flooded throughout the AS (except for stub areas).
AS path	In BGP, the route to a destination. The path consists of the AS numbers of all routers a packet must go through to reach a destination.
ASIC	Application-specific integrated circuit. Specialized processors that perform specific functions on the router.
ATM	Asynchronous Transfer Mode. A high-speed multiplexing and switching method utilizing fixed-length cells of 53 octets to support multiple types of traffic.
atomic	Smallest possible operation. An atomic operation is performed either entirely or not at all. For example, if machine failure prevents a transaction from completing, the system is rolled back to the start of the transaction, with no changes taking place.
Automatic Protection Switching	See APS.
autonomous system	See AS.
autonomous system boundary router	In OSPF, routers that exchange routing information with routers in other ASs.
autonomous system external link advertisements	OSPF link-state advertisement sent by autonomous system boundary routers to describe external routes that they know. These link-state advertisements are flooded throughout the autonomous system (except for stub areas).
autonomous system path	In BGP, the route to a destination. The path consists of the autonomous system numbers of all the routers a packet must pass through to reach a destination.
B	
backbone area	In OSPF, an area that consists of all networks in area ID 0.0.0.0, their attached routers, and all area border routers.
backplane	On an M40 router, component of the Packet Forwarding Engine that distributes power, provides signal connectivity, manages shared memory on FPCs, and passes outgoing data cells to FPCs.
bandwidth	The range of transmission frequencies a network can use, expressed as the difference between the highest and lowest frequencies of a transmission channel. In computer networks, greater bandwidth indicates faster data-transfer rate capacity.
Bellcore	Bell Communications Research. Research and development organization created after the divestiture of the Bell System. It is supported by the regional Bell holding companies (RBHCs), which own the regional Bell operating companies (RBOCs).
BERT	Bit error rate test. A test that can be run on a T3 interface to determine whether it is operating properly.
BGP	Border Gateway Protocol. Exterior gateway protocol used to exchange routing information among routers in different autonomous systems.
bit error rate test	See <i>BERT</i> .

	BITS	Building Integrated Timing Source. Dedicated timing source that synchronizes all equipment in a particular building.
	Border Gateway Protocol	See BGP.
	broadcast	Operation of sending network traffic from one network node to all other network nodes.
	bundle	Collection of software that makes up a JUNOS software release.
C	CCC	Circuit cross-connect. A JUNOS software feature that allows you to configure transparent connections between two circuits, where a circuit can be a Frame Relay DLCI, an ATM VC, a PPP interface, a Cisco HDLC interface, or an MPLS label-switched path (LSP).
	CE device	Customer edge device. Router or switch in the customer's network that is connected to a service provider's provider edge (PE) router and participates in a Layer 3 VPN.
	CFM	Cubic feet per minute. Measure of fan speed.
	channel service unit	See <i>CSU/DSU</i> .
	CIDR	Classless interdomain routing. A method of specifying Internet addresses in which you explicitly specify the bits of the address to represent the network address instead of determining this information from the first octet of the address.
	CIP	Connector Interface Panel. On an M160 router, the panel that contains connectors for the Routing Engines, BITS interfaces, and alarm relay contacts.
	circuit cross-connect	See <i>CCC</i> .
	class of service	See <i>CoS</i> .
	CLEC	(Pronounced "see-lek") Competitive Local Exchange Carrier. Company that competes with the already established local telecommunications business by providing its own network and switching.
	CLEI	Common language equipment identifier. Inventory code used to identify and track telecommunications equipment.
	CLI	Command-line interface. Interface provided for configuring and monitoring the routing protocol software.
	client peer	In a BGP route reflection, a member of a cluster that is not the route reflector. See also <i>nonclient peer</i> .
	CLNP	Connectionless Network Protocol. ISO-developed protocol for OSI connectionless network service. CLNP is the OSI equivalent of IP.
	cluster	In BGP, a set of routers that have been grouped together. A cluster consists of one system that acts as a route reflector, along with any number of client peers. The client peers receive their route information only from the route reflector system. Routers in a cluster do not need to be fully meshed.

community	In BGP, a group of destinations that share a common property. Community information is included as one of the path attributes in BGP update messages.
confederation	In BGP, a group of systems that appears to external autonomous systems to be a single autonomous system.
constrained path	In traffic engineering, a path determined using RSVP signaling and constrained using CSPF. The ERO carried in the packets contains the constrained path information.
core	The central backbone of the network.
CoS	Class of service. A group of privileges and features assigned to a particular service.
CPE	Customer premises equipment. Telephone or other service provider equipment located at a customer site.
craft interface	Mechanisms used by a Communication Workers of America craftsman to operate, administer, and maintain equipment or provision data communications. On a Juniper Networks router, the craft interface allows you to view status and troubleshooting information and perform system control functions.
CSNP	Complete sequence number PDU. Packet that contains a complete list of all the LSPs in the IS-IS database.
CSPF	Constrained Shortest Path First. An MPLS algorithm that has been modified to take into account specific restrictions when calculating the shortest path across the network.
CSU/DSU	Channel service unit/data service unit. Channel service unit connects a digital phone line to a multiplexer or other digital signal device. Data service unit connects a DTE to a digital phone line.
customer edge device	See CE device.
D	
daemon	Background process that performs operations on behalf of the system software and hardware. Daemons normally start when the system software is booted, and they run as long as the software is running. In the JUNOS software, daemons are also referred to as processes.
damping	Method of reducing the number of update messages sent between BGP peers, thereby reducing the load on these peers without adversely affecting the route convergence time for stable routes.
data circuit-terminating equipment	See DCE.
data-link connection identifier	See DLCI.
data service unit	See CSU/DSU.
Data Terminal Equipment	See DTE.
dcd	The JUNOS software interface process (daemon).
DCE	Data circuit-terminating equipment. RS-232-C device, typically used for a modem or printer, or a network access and packet switching node.

default address	Router address that is used as the source address on unnumbered interfaces.
denial of service	<i>See DoS.</i>
dense wavelength-division multiplexing	<i>See DWDM.</i>
designated router	In OSPF, a router selected by other routers that is responsible for sending link-state advertisements that describe the network, which reduces the amount of network traffic and the size of the routers' topological databases.
destination prefix length	Number of bits of the network address used for host portion of a CIDR IP address.
DHCP	Dynamic Host Configuration Protocol. Allocates IP addresses dynamically so that they can be reused when they are no longer needed.
Dijkstra algorithm	<i>See SPF.</i>
DIMM	Dual inline memory module. 168-pin memory module that supports 64-bit data transfer.
direct routes	<i>See interface routes.</i>
DLCI	Data-link connection identifier. Identifier for a Frame Relay virtual connection (also called a logical interface).
DoS	Denial of service. System security breach in which network services become unavailable to users.
DRAM	Dynamic random-access memory. Storage source on the router that can be accessed quickly by a process.
drop profile	Drop probabilities for different levels of buffer fullness that are used by RED to determine from which queue to drop packets.
DSU	Data service unit. A device used to connect a DTE to a digital phone line. Converts digital data from a router to voltages and encoding required by the phone line. <i>See also CSU/DSU.</i>
DTE	Data Terminal Equipment. RS-232-C interface that a computer uses to exchange information with a serial device.
DVMRP	Distance Vector Multicast Routing Protocol. Distributed multicast routing protocol that dynamically generates IP multicast delivery trees using a technique called reverse path multicasting (RPM) to forward multicast traffic to downstream interfaces.
DWDM	Dense wavelength-division multiplexing. Technology that enables data from different sources to be carried together on an optical fiber, with each signal carried on its own separate wavelength.
Dynamic Host Configuration Protocol	<i>See DHCP.</i>

E

- EBGP** External BGP. BGP configuration in which sessions are established between routers in different ASs.
- ECSA** Exchange Carriers Standards Association. A standards organization created after the divestiture of the Bell System to represent the interests of interexchange carriers.
- edge router** In MPLS, a router located at the beginning or end of a label-switching tunnel. When at the beginning of a tunnel, an edge router applies labels to new packets entering the tunnel. When at the end of a tunnel, the edge router removes labels from packets exiting the tunnel. See also *MPLS*.
- EGP** Exterior gateway protocol, such as BGP.
- egress router** In MPLS, last router in a label-switched path (LSP). See also *ingress router*.
- EIA** Electronic Industries Association. A United States trade group that represents manufacturers of electronics devices and sets standards and specifications.
- EMI** Electromagnetic interference. Any electromagnetic disturbance that interrupts, obstructs, or otherwise degrades or limits the effective performance of electronics or electrical equipment.
- end system** In IS-IS, network entity that sends and receives packets.
- ERO** Explicit Route Object. Extension to RSVP that allows an RSVP PATH message to traverse an explicit sequence of routers that is independent of conventional shortest-path IP routing.
- explicit path** See *signaled path*.
- Explicit Route Object** See *ERO*.
- export** To place routes from the routing table into a routing protocol.
- external BGP** See *EBGP*.
- external metric** A cost included in a route when OSPF exports route information from external autonomous systems. There are two types of external metrics: Type 1 and Type 2. Type 1 external metrics are equivalent to the link-state metric; that is, the cost of the route, used in the internal autonomous system. Type 2 external metrics are greater than the cost of any path internal to the autonomous system.

F

- fast reroute** Mechanism for automatically rerouting traffic on an LSP if a node or link in an LSP fails, thus reducing the loss of packets traveling over the LSP.
- FEAC** Far-end alarm and control. T3 signal used to send alarm or status information from the far-end terminal back to the near-end terminal and to initiate T3 loopbacks at the far-end terminal from the near-end terminal.
- FEB** Forwarding Engine Board. In M5 and M10 routers, provides route lookup, filtering, and switching to the destination port.
- flap damping** See *damping*.
- flapping** See *route flapping*.

Flexible PIC Concentrator	See FPC.
Forwarding Engine Board	See FEB.
forwarding information base	See forwarding table.
forwarding table	JUNOS software forwarding information base (FIB). The JUNOS routing protocol process installs active routes from its routing tables into the Routing Engine forwarding table. The kernel copies this forwarding table into the Packet Forwarding Engine, which is responsible for determining which interface transmits the packets.
FPC	Flexible PIC Concentrator. An interface concentrator on which PICs are mounted. An FPC inserts into a slot in a Juniper Networks router. <i>See also PIC.</i>
FRU	Field-replaceable unit. Router component that customers can replace onsite.
G	
group	A collection of related BGP peers.
H	
HDLC	High-level data link control. An International Telecommunication Union (ITU) standard for a bit-oriented data link layer protocol on which most other bit-oriented protocols are based.
hold time	Maximum number of seconds allowed to elapse between the time a BGP system receives successive keepalive or update messages from a peer.
host module	On an M160 router, provides routing and system management functions of the router. Consists of the Routing Engine and Miscellaneous Control Subsystem (MCS).
I	
IANA	Internet Assigned Numbers Authority. Regulatory group that maintains all assigned and registered Internet numbers, such as IP and multicast addresses. <i>See also NIC.</i>
IBGP	Internal BGP. BGP configuration in which sessions are established between routers in the same ASs.
ICMP	Internet Control Message Protocol. Used in router discovery, ICMP allows router advertisements that enable a host to discover addresses of operating routers on the subnet.
IDE	Integrated Drive Electronics. Type of hard disk on the Routing Engine.
IEC	International Electrotechnical Commission. <i>See ISO.</i>
IEEE	Institute of Electronic and Electrical Engineers. International professional society for electrical engineers.
IETF	Internet Engineering Task Force. International community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.
IGMP	Internet Group Membership Protocol. Used with multicast protocols to determine whether group members are present.

IGP	Interior gateway protocol, such as IS-IS, OSPF, and RIP.
import	To install routes from the routing protocols into a routing table.
ingress router	In MPLS, first router in a label-switched path (LSP). <i>See also egress router.</i>
inter-AS routing	Routing of packets among different ASs. <i>See also EBGp.</i>
intercluster reflection	In a BGP route reflection, the redistribution of routing information by a route reflector system to all nonclient peers (BGP peers not in the cluster). <i>See also route reflection.</i>
interface routes	Routes that are in the routing table because an interface has been configured with an IP address. Also called <i>direct routes</i> .
intermediate system	In IS-IS, network entity that sends and receives packets and that can also route packets.
internal BGP	<i>See IBGP.</i>
intra-AS routing	The routing of packets within a single AS. <i>See also IBGP.</i>
IP	Internet Protocol. The protocol used for sending data from one point to another on the Internet.
IS-IS	Intermediate System-to-Intermediate System protocol. Link-state, interior gateway routing protocol for IP networks that also uses the shortest-path first (SPF) algorithm to determine routes.
ISO	International Organization for Standardization. Worldwide federation of standards bodies that promotes international standardization and publishes international agreements as International Standards.
ISP	Internet service provider. Company that provides access to the Internet and related services.
ITU	International Telecommunications Union (formerly known as the CCITT). Group supported by the United Nations that makes recommendations and coordinates the development of telecommunications standards for the entire world.
J	
jitter	Small random variation introduced into the value of a timer to prevent multiple timer expirations from becoming synchronized.
K	
kernel forwarding table	<i>See forwarding table.</i>
L	
label	In MPLS, 20-bit unsigned integer in the range 0 through 1048575, used to identify a packet traveling along an LSP.
label-switched path (LSP)	Sequence of routers that cooperatively perform MPLS operations for a packet stream. The first router in an LSP is called the <i>ingress router</i> ; and the last router in the path is called the <i>egress router</i> . An LSP is a point-to-point, half-duplex connection from the ingress router to the egress router. (The ingress and egress routers cannot be the same router.)
label switching	<i>See MPLS.</i>
label-switching router	<i>See LSR.</i>

link	Communication path between two neighbors. A link is <i>up</i> when communication is possible between the two end points.
link-state PDU (LSP)	Packets that contain information about the state of adjacencies to neighboring systems.
local preference	Optional BGP path attribute carried in internal BGP update packets that indicates the degree of preference for an external route.
loose	In the context of traffic engineering, a path that can use any route or any number of other intermediate (transit) points to reach the next address in the path. (Definition from RFC 791, modified to fit LSPs.)
LSP	See label-switched path (LSP) and link-state PDU (LSP).
LSR	Label-switching router. A router on which MPLS and RSVP are enabled and is thus capable of processing label-switched packets.
M	
martian address	Network address about which all information is ignored.
mask	<i>See subnet mask.</i>
MBGP	Multiprotocol BGP. An extension to BGP that allows you to connect multicast topologies within and between BGP ASs.
MBone	Internet multicast backbone. An interconnected set of subnetworks and routers that support the delivery of IP multicast traffic. The MBone is a virtual network that is layered on top of sections of the physical Internet.
MCS	Miscellaneous Control Subsystem. On an M160 router, provides control and monitoring functions for router components and SONET clocking for the router.
MED	Multiple exit discriminator. Optional BGP path attribute consisting of a metric value that is used to determine the exit point to a destination when all other factors in determining the exit point are equal.
mesh	Network topology in which devices are organized in a manageable, segmented manner with many, often redundant, interconnections between network nodes.
MIB	Management Information Base. Definition of an object that can be managed by SNMP.
midplane	Forms the rear of the PIC cage on M5 and M10 routers and the FPC card cage on M20 and M160 routers. Provides data transfer, power distribution, and signal connectivity.
Miscellaneous Control Subsystem	<i>See MCS.</i>
MPLS	Multiprotocol Label Switching. Mechanism for engineering network traffic patterns that functions by assigning to network packets short labels that describe how to forward them through the network. Also called <i>label switching</i> . <i>See also traffic engineering.</i>
MTBF	Mean time between failure. Measure of hardware component reliability.

MTU	Maximum transfer unit. Limit on segment size for a network.
multicast	Operation of sending network traffic from one network node to multiple network nodes.
multiprotocol BGP	See MBGP.
Multiprotocol Label Switching	See MPLS.
N	
neighbor	Adjacent system reachable by traversing a single subnetwork. An immediately adjacent router. Also called a <i>peer</i> .
NET	Network entity title. Network address defined by the ISO network architecture and used in CLNS-based networks.
network layer reachability information	See <i>NLRI</i> .
network link advertisement	An OSPF link-state advertisement flooded throughout a single area by designated routers to describe all routers attached to the network.
Network Time Protocol	See <i>NTP</i> .
NIC	Network Information Center. Internet authority responsible for assigning Internet-related numbers, such as IP addresses and autonomous system numbers. See also <i>IANA</i> .
NLRI	Network layer reachability information. Information that is carried in BGP packets and is used by MBGP.
nonclient peer	In a BGP route reflection, a BGP peer that is not a member of a cluster. See also <i>client peer</i> .
not-so-stubby area	See <i>NSSA</i> .
NSAP	Network service access point. Connection to a network that is identified by a network address.
n-selector	Last byte of an nonclient peer address.
NSSA	Not-so-stubby area. In OSPF, a type of stub area in which external routes can be flooded.
NTP	Network Time Protocol. Protocol used to synchronize computer clock times on a network.
O	
OC	Optical Carrier. In SONET, Optical Carrier levels indicate the transmission rate of digital signals on optical fiber.
OSI	Open System Interconnection. Standard reference model for how messages are transmitted between two points on a network.
OSPF	Open Shortest Path First. A link-state IGP that makes routing decisions based on the shortest-path-first (SPF) algorithm (also referred to as the <i>Dijkstra algorithm</i>).

P

package	A collection of files that make up a JUNOS software component.
Packet Forwarding Engine	The architectural portion of the router that processes packets by forwarding them between input and output interfaces.
path attribute	Information about a BGP route, such as the route origin, AS path, and next-hop router.
PCI	Peripheral Component Interconnect. Standard, high-speed bus for connecting computer peripherals. Used on the Routing Engine.
PCMCIA	Personal Computer Memory Card International Association. Industry group that promotes standards for credit card-size memory or I/O devices.
PDU	Protocol data unit. IS-IS packets.
PE router	Provider edge router. A router in the service provider's network that is connected to a customer edge (CE) device and that participates in a Virtual Private Network (VPN).
PEC	Policing Equivalence Classes. In traffic policing, a set of packets that is treated the same by the packet classifier.
peer	An immediately adjacent router with which a protocol relationship has been established. Also called a <i>neighbor</i> .
PFE	<i>See Packet Forwarding Engine.</i>
Physical Interface Card	<i>See PIC.</i>
PIC	Physical Interface Card. A network interface-specific card that can be installed on an FPC in the router.
PIM	Protocol Independent Multicast. A protocol-independent multicast routing protocol. PIM Sparse Mode routes to multicast groups that might span wide-area and interdomain internets. PIM Dense Mode is a flood-and-prune protocol.
PLP	Packet Loss Priority.
policing	Applying rate limits on bandwidth and burst size for traffic on a particular interface.
pop	Removal of the last label, by a router, from a packet as it exits an MPLS domain.
PPP	Point-to-Point Protocol. Link-layer protocol that provides multiprotocol encapsulation. It is used for link-layer and network-layer configuration.
preference	Desirability of a route to become the active route. A route with a lower preference value is more likely to become the active route. The preference is an arbitrary value in the range 0 through 255 that the routing protocol process uses to rank routes received from different protocols, interfaces, or remote systems.
preferred address	On an interface, the default local address used for packets sourced by the local router to destinations on the subnet.
primary address	On an interface, the address used by default as the local address for broadcast and multicast packets sourced locally and sent out the interface.

primary interface	Router interface that packets go out when no interface name is specified and when the destination address does not imply a particular outgoing interface.
Protocol-Independent Multicast	See PIM.
provider edge router	See <i>PE router</i> .
provider router	Router in the service provider's network that does not attach to a customer edge (CE) device.
PSNP	Partial sequence number PDU. Packet that contains only a partial list of the LSPs in the IS-IS link-state database.
push	Addition of a label or stack of labels, by a router, to a packet as it enters an MPLS domain.
Q	
QoS	Quality of service. Performance, such as transmission rates and error rates, of a communications channel or system.
quality of service	See <i>QoS</i> .
R	
RADIUS	Remote Authentication Dial-In User Service. Authentication method for validating users who attempt to access the router using Telnet.
Random Early Detection	See <i>RED</i> .
rate limiting	See <i>policing</i> .
RBOC	(Pronounced "are-bock") Regional Bell operating company. Regional telephone companies formed as a result of the divestiture of the Bell System.
RED	(Pronounced "red") Random Early Detection. Gradual drop profile for a given class that is used for congestion avoidance. RED tries to anticipate incipient congestion and reacts by dropping a small percentage of packets from the head of the queue to ensure that a queue never actually becomes congested.
Resource Reservation Protocol	See <i>RSVP</i> .
RFC	Request for Comments. Internet standard specifications published by the Internet Engineering Task Force.
RFI	Radio frequency interference. Interference from high-frequency electromagnetic waves emanating from electronic devices.
RIP	Routing Information Protocol. Distance-vector interior gateway protocol that makes routing decisions based on hop count.
route flapping	Situation in which BGP systems send an excessive number of update messages to advertise network reachability information.
route identifier	IP address of the router from which a BGP, IGP, or OSPF packet originated.

route reflection	In BGP, configuring a group of routers into a cluster and having one system act as a route reflector, redistributing routes from outside the cluster to all routers in the cluster. Routers in a cluster do not need to be fully meshed.
router link advertisement	OSPF link-state advertisement flooded throughout a single area by all routers to describe the state and cost of the router's links to the area.
routing domain	See AS.
Routing Engine	Architectural portion of the router that handles all routing protocol processes, as well as other software processes that control the router's interfaces, some of the chassis components, system management, and user access to the router.
routing table	Common database of routes learned from one or more routing protocols. All routes are maintained by the JUNOS routing protocol process.
rp	JUNOS software routing protocol process (daemon). User-level background process responsible for starting, managing, and stopping the routing protocols on a Juniper Networks router.
RPM	Reverse path multicasting. Routing algorithm used by DVMRP to forward multicast traffic.
RSVP	Resource Reservation Protocol. Resource reservation setup protocol designed to interact with integrated services on the Internet.
S	SAP Session Announcement Protocol. Used with multicast protocols to handle session conference announcements.
	SAR Segmentation and reassembly. Buffering used with ATM.
	SCB System Control Board. On an M40 router, the part of the Packet Forwarding Engine that performs route lookups, monitors system components, and controls FPC resets.
	SDH Synchronous Digital Hierarchy. CCITT variation of SONET standard.
	SDP Session Description Protocol. Used with multicast protocols to handle session conference announcements.
	SDRAM Synchronous Dynamic Random Access Memory.
	secure shell See <i>SSH</i> .
	SFM Switching and Forwarding Module. On an M160 router, a component of the Packet Forwarding Engine that provides route lookup, filtering, and switching to FPCs.
	shortest-path-first algorithm See <i>SPF</i> .
	signaled path In traffic engineering, an explicit path; that is, a path determined using RSVP signaling. The ERO carried in the packets contains the explicit path information.
simplex interface	An interface that assumes that packets it receives from itself are the result of a software loopback process. The interface does not consider these packets when determining whether the interface is functional.

	SNMP	Simple Network Management Protocol. Protocol governing network management and the monitoring of network devices and their functions.
	SONET	Synchronous Optical Network. High-speed (up to 2.5 Gbps) synchronous network specification developed by Bellcore and designed to run on optical fiber. STS-1 is the basic building block of SONET. Approved as an international standard in 1988. <i>See also SDH.</i>
	SPF	Shortest-path first, an algorithm used by IS-IS and OSPF to make routing decisions based on the state of network links. Also called the <i>Dijkstra algorithm</i> .
	SSB	System and Switch Board. On an M20 router, Packet Forwarding Engine component that performs route lookups and component monitoring and monitors FPC operation.
	SSH	Secure shell. Software that provides a secured method of logging in to a remote network system.
	SSRAM	Synchronous Static Random Access Memory.
	static LSP	<i>See static path.</i>
	static path	In the context of traffic engineering, a static route that requires hop-by-hop manual configuration. No signaling is used to create or maintain the path. Also called a <i>static LSP</i> .
	STM	Synchronous Transport Module. CCITT specification for SONET at 155.52 Mbps.
	strict	In the context of traffic engineering, a route that must go directly to the next address in the path. (Definition from RFC 791, modified to fit LSPs.)
	STS	Synchronous Transport Signal. Synchronous Transport Signal level 1. Basic building block signal of SONET, operating at 51.84 Mbps. Faster SONET rates are defined as STS- <i>n</i> , where <i>n</i> is a multiple of 51.84 Mbps. <i>See also SONET.</i>
	stub area	In OSPF, an area through which, or into which, AS external advertisements are not flooded.
	subnet mask	Number of bits of the network address used for host portion of a Class A, Class B, or Class C IP address.
	summary link advertisement	OSPF link-statement advertisement flooded throughout the advertisement's associated areas by area border routers to describe the routes that they know about in other areas.
	sysid	System identifier. Portion of the ISO nonclient peer. The sysid can be any 6 bytes that are unique throughout a domain.
	System and Switch Board	<i>See SSB.</i>
T	TACACS+	Terminal Access Controller Access Control System Plus. Authentication method for validating users who attempt to access the router using Telnet.
	TCP	Transmission Control Protocol. Works in conjunction with Internet Protocol (IP) to send data over the Internet. Divides a message into packets and tracks the packets from point of origin to destination.
	ToS	Type of service.

traffic engineering Process of selecting the paths chosen by data traffic in order to balance the traffic load on the various links, routers, and switches in the network. (Definition from <http://www.ietf.org/internet-drafts/draft-ietf-mpls-framework-04.txt>.) *See also MPLS.*

transit area In OSPF, an area used to pass traffic from one adjacent area to the backbone or to another area if the backbone is more than two hops away from an area.

transit router In MPLS, any intermediate router in the LSP between the ingress router and the egress router.

tunnel Private, secure path through an otherwise public network.

type of service *See ToS.*

U

unicast Operation of sending network traffic from one network node to another individual network node.

UPS Uninterruptible power supply. Device that sits between a power supply and a router (or other piece of equipment) the prevents undesired power-source events, such as outages and surges, from affecting or damaging the device.

V

VCI Virtual circuit identifier. Identifier for an ATM virtual connection. Also called a *logical interface*.

virtual circuit identifier *See VCI.*

virtual link In OSPF, a link created between two routers that are part of the backbone but are not physically contiguous.

virtual path identifier Virtual circuit identifier. *See VCI.*

Virtual Router Redundancy Protocol *See VRRP.*

VPI *See VCI.*

VRRP Virtual Router Redundancy Protocol. On Fast Ethernet and Gigabit Ethernet interfaces, allows you to configure virtual default routers.

W

wavelength-division multiplexing *See WDM.*

WDM Wavelength-division multiplexing. Technique for transmitting a mix of voice, data, and video over various wavelengths (colors) of light.

weighted round-robin *See WRR.*

WRR Weighted round-robin. Scheme used to decide the queue from which the next packet should be transmitted.

Part 8

Indexes

- Index on page 337
- Index of Statements and Commands on page 347

.....

Index

Index

Bold numbers point to command summary pages.

Symbols

#, in configuration statements.....	xxvi
(), in syntax descriptions	xxv
< > , in syntax descriptions.....	xxv
[], in configuration statements.....	xxv
{ }, in configuration statements.....	xxv
(pipe), in syntax descriptions	xxv

A

active option.....	52
adaptive rerouting	61, 85
adaptive statement	85
usage guidelines	61
address (tracing flag)	154, 161, 293
addresses	
associating with LSPs	52, 92
egress router address	48, 108
ingress router address	48, 90
admin-group statement	85
usage guidelines	58
admin-groups statement.....	86
usage guidelines	57
administrative groups.....	52, 57, 85, 86, 88, 91
advertise-hold-time statement.....	86
advertisement messages, LDP	143
advertisement-hold-time statement	
usage guidelines	65
advertising transitions	65, 86
aggregate statement.....	129
usage guidelines	122
aggregation, RSVP	122, 129
all (tracing flag).....	109, 135
allocation of labels.....	22
ATM circuits	170, 175
authentication, RSVP	123, 130
authentication-key statement.....	130
usage guidelines	123

B

backup paths	28
<i>See also</i> fate sharing	
bandwidth	
LSP paths	64
RSVP reservations	124, 134
bandwidth statement	
fast reroute	87
usage guidelines	52
RSVP	130
signaled LSPs	87
usage guidelines	64
BGP destinations.....	33
binding (tracing flag).....	154, 161, 293
braces, in configuration statements.....	xxv
brackets	
angle, in syntax descriptions	xxv
square, in configuration statements	xxv

C

calculating LSPs <i>See</i> path calculation	
CCC	
example configurations	172, 176, 178
Layer 2 switching cross-connects.....	167, 169, 179, 180
LSP stitching cross-connects.....	167, 177
MPLS tunneling cross-connects	167, 173, 181
overview.....	167
CCC connections	
Layer 2 switching cross-connects	171, 179, 180
LSP stitching cross-connects.....	177
MPLS tunneling cross-connects	175, 181
CCC encapsulation	
Layer 2 switching cross-connects	170
MPLS tunneling cross-connects	174
CCC encapsulation types, Layer 2 VPNs.....	290, 311
circuit cross-connect <i>See</i> CCC	
Cisco HDLC circuits	170, 175

- class-of-service statement
 - ingress router configuration **87**
 - usage guidelines 74
 - signaled LSPs **87**
 - usage guidelines 60
 - static LSPs **87**
 - usage guidelines 76
- colored links 52, 57, 86
- comments, in configuration statements xxvi
- computing LSPs *See* path calculation
- connection (tracing flag) 83, 109
- connection-detail (tracing flag) 83, 109
- connections statement **179**
 - usage guidelines 10, 171
- constrained path computation 56, 97
- Constrained Shortest Path First algorithm *See* CSPF algorithm
- constrained-path LSPs 24, 25, 68
 - *See also* CSPF algorithm
- conventions, documentation xxiv
- CoS values 60
- cross-connect, circuit *See* CCC
- cspf (tracing flag) 83, 109
- CSPF algorithm
 - fate sharing 67
 - offline path computation 6, 27
 - online path computation 25, 56, 97
 - overview 5
- cspf-link (tracing flag) 83, 109
- cspf-node (tracing flag) 83, 109
- curly braces, in configuration statements xxv
- customer support, contacting xxvi

D

- damping LSP transitions 65
- detail (tracing flag modifier) 109, 135, 162, 315
- detours *See* fast reroute
- disable statement
 - LDP **157**
 - usage guidelines 146
 - MPLS **88**
 - usage guidelines 56
 - RSVP **131**
 - usage guidelines 122
- discard statement **88**
 - usage guidelines 76
- discovery messages, LDP 142
- distinct reservations 119
- documentation conventions xxiv
- dynamic LSP metric 53

E

- egress routers
 - example configuration 77
 - overview 24
 - signaled LSPs 48
 - static LSPs 75, 92
- empty paths 44, 100
- encapsulation statement
 - usage guidelines 170, 175
- encapsulation types, Layer 2 VPNs 290, 292, 311, 312
- encapsulation-type statement **311, 312**
- error (tracing flag) 83, 109, 125, 134, 154, 161, 293
- event (tracing flag) 154, 161, 293
- exclude statement **88**
 - usage guidelines 52, 59
- Explicit Null Label 21
- Explicit Route Object 6
- explicit routes 5
- explicit senders, RSVP 119
- explicit-path LSPs 24, 25, 68, 79

F

- failed LSPs
 - fast reroute 38, 49, 87, 89
 - standby secondary paths 38
- family inet-vpn option 206
- family l2vpn option 288
- family mpls statement, usage guidelines 140
- fast reroute
 - configuring 51, 87, 89
 - overview 38, 49
- fast-reroute statement **89**
 - usage guidelines 51
- fate sharing
 - CSPF algorithm 67
 - example configuration 67
 - fate-sharing groups 66
 - overview 28
 - signaled LSPs 66, 89
- fate-sharing groups 66
- fate-sharing statement **89**
 - usage guidelines 66
- FECs 139
- FF (reservation style) 119
- files, MPLS statistics output 81
- filtering received labels 140, 148, 159
- fixed filter reservation style 119
- forwarding
 - *See* MPLS
- Forwarding Equivalence Classes 139
- forwarding next hop 35
- Frame Relay circuits 171, 175
- from statement **90**
 - usage guidelines 48, 148

- G**
- general (tracing flag) 109, 135
 - group-name statement, usage guidelines 57
 - groups
 - administrative groups 52, 57, 85, 86, 88, 91
 - fate-sharing groups 66
- H**
- hello interval
 - LDP 147, 158
 - RSVP 123, 131
 - hello-interval statement
 - LDP **158**
 - usage guidelines 147
 - RSVP **131**
 - usage guidelines 123
 - hold priority 62
 - hold time
 - LDP 147, 159
 - signaled LSPs 65, 86
 - hold-time statement **159**
 - usage guidelines 147
 - hop-limit statement **91**
 - usage guidelines 64
 - host routes 33, 52
 - hot-standby state 64
- I**
- IGP destinations 35
 - IGP shortcuts
 - enabling 30
 - LSP metrics 54
 - overview 28
 - qualified LSPs 30
 - routing tables 31
 - uses of shortcuts 30
 - IGPs
 - advertising LSPs 32
 - configuring on Layer 2 VPNs 288
 - Implicit Null Label 21
 - import statement **159**
 - usage guidelines 140, 148
 - include statement **91**
 - usage guidelines 52, 59
 - inet option 74
 - inet.0 routing table
 - IGP shortcuts 31
 - MPLS 36
 - inet.3 routing table
 - IGP shortcuts 31
 - installing routes 52
 - MPLS 36
 - information distribution component, traffic engineering .
5
 - ingress routers
 - configuring for static LSPs 73, 87, 96, 106
 - example configurations 68, 74
 - overview 24
 - path connection retry information 53, 104
 - initialization (tracing flag) 154, 161, 293
 - install statement **92**
 - usage guidelines 52
 - instance-type statement **279, 312**
 - l2vpn option
 - usage guidelines 289
 - vrf option
 - usage guidelines 207
 - Integrity Object 116
 - interface (from operator, LDP) 148
 - interface statement **279, 313**
 - LDP **159**
 - usage guidelines 146
 - RSVP **131**
 - usage guidelines 122
 - static LSPs **92**
 - usage guidelines 75
 - VPNs
 - usage guidelines 207
 - interface-switch statement **180**
 - usage guidelines 171
 - intermediate routers
 - configuring for static LSPs 75, 92
 - example configurations 77
 - intra-region LSPs 31
 - IPv4 Explicit Null Label 21
 - IPv6 Implicit Null Label 21
- K**
- keep multiplier, RSVP 124, 132
 - keepalive interval 147, 160
 - keepalive timeout 147, 160
 - keepalive-interval statement **160**
 - usage guidelines 147
 - keepalive-timeout statement **160**
 - usage guidelines 147
 - keep-multiplier statement **132**
 - usage guidelines 125
- L**
- label (tracing flag) 154, 161, 293
 - Label Distribution Protocol *See* LDP
 - label filtering 140, 148, 159
 - Label Object 6
 - label operations, LDP 141
 - label properties 74
 - Label Request Object 6
 - label stacks 22
 - label swapping 4

- label-map statement.....**93**
 - usage guidelines.....76
- labels
 - label allocation.....22
 - label operations.....23
 - numerical ranges.....21
 - overview.....19
 - reserved labels.....21
- label-switched paths *See* LSPs
- label-switched-path statement.....**94**
 - usage guidelines.....45, 127
- Layer 2 switching cross-connects
 - CCC connections.....171, 179, 180
 - CCC encapsulation.....170
 - configuring MPLS.....171
 - example configuration.....172
 - overview.....167, 169
- Layer 2 VPNs
 - CCC encapsulation types.....290, 311
 - configuration example.....295
 - configuring.....289, 313
 - encapsulation types.....290, 292, 311, 312
 - IBGP session between PE routers.....288
 - IGP, configuring.....288
 - instance types.....312
 - LDP, enabling.....285
 - MPLS LSPs, configuring.....284
 - route distinguisher.....208
 - route distinguishers.....291, 313
 - routing instances.....289
 - RSVP.....286
 - site configuration.....292, 314
 - tracing protocol traffic.....314
- Layer 3 VPNs
 - example configurations.....227
- LDP
 - configuration statements.....145, 157
 - configuring.....146, 159, 160
 - disabling.....88, 146, 157
 - enabling.....146
 - enabling for Layer 2 VPNs.....285
 - enabling for VPNs.....203
 - example configurations.....149, 155
 - hello interval.....147, 158
 - hold time.....147, 159
 - JUNOS implementation.....140
 - keepalive interval.....147, 160
 - keepalive timeout.....147, 160
 - label operations.....141
 - message types.....142
 - operations.....140
 - overview.....139
 - received label filtering.....140, 148, 159
 - route preferences.....148, 161
 - standards.....140
 - tracing protocol traffic.....154, 161, 293
 - tunneling through RSVP LSPs.....95, 141, 152
- ldp statement.....**160**
 - usage guidelines.....11, 146
- LDP tunneling.....65
- ldp-tunneling statement.....**95**
 - usage guidelines.....152
- least fill tie-breaking rule.....27, 54, 103
- least-fill statement.....**103**
- link attributes, in CSPF algorithm.....26
- link coloring.....52, 57, 86
- link-layer protocols.....20
- load balancing
 - load balancing without CSPF.....55
 - per-prefix load balancing.....38
- log-updown statement.....**95**
 - usage guidelines.....82
- loose explicit routes.....5, 79
- loose option.....44
- LSP attributes, in CSPF algorithm.....25
- LSP stitching cross-connects.....167, 177
- LSPs
 - adaptive rerouting.....61, 85
 - administrative groups.....52, 57, 85, 86, 88, 91
 - advertising in IGPs.....32
 - associating addresses.....52, 92
 - configuration statements.....45, 94
 - constrained path computation.....56, 97
 - constrained-path LSPs.....24, 25
 - CoS values.....60
 - creating.....45
 - damping LSP transitions.....65
 - egress routers.....48, 75, 77, 92
 - example configurations.....68
 - explicit-path LSPs.....24, 25, 79
 - fast reroute.....38, 49, 87, 89
 - fate sharing.....28, 66, 89
 - forwarding next hop, selecting.....35
 - hold time.....65, 86
 - host routes.....33
 - IGP shortcuts.....28
 - ingress routers.....48, 90
 - intermediate routers.....75, 92
 - intraregion LSPs.....31
 - LDP tunneling.....65
 - load balancing without CSPF.....55
 - LSP failure.....38
 - metrics.....53, 54, 96
 - MPLS routers, configuring.....68
 - named paths.....44, 100
 - names.....46
 - overview.....4, 20
 - packet traversal.....4, 24
 - path bandwidth.....64
 - path calculation *See* path calculation
 - path connection retry information.....53, 104
 - path length.....64, 91
 - per-prefix load balancing.....38
 - preemption.....62, 102

preference levels 59, 101
 primary LSPs 49, 101
 priorities 62, 102
 recording routes 59
 reoptimization 63, 99
 router functions 24
 routing options 6
 RSVP *See* RSVP
 scope of LSPs 25
 secondary LSPs 49, 105
 signaled *See* signaled LSPs
 standby secondary paths 38
 standby state 64, 105
 static *See* static LSPs
 tie-breaking rules 27, 54, 103
 traffic engineering, configuring 81
 TTL decrementing, disabling 55, 97, 98
 tunneling through RSVP LSPs 95, 141, 152
See also labels; LDP
 lsp-switch statement **180**
 usage guidelines 177

M
 MD5 authentication 123
 messages
 LDP message types 142
 MPLS syslog messages 82, 95
 RSVP message types 117
 RSVP refresh messages 124
 metric statement **96**
 usage guidelines 54
 metrics
 dynamic LSP metric 53
 static LSP metric 54, 96
 most fill tie-breaking rule 27, 55, 103
 most-fill statement **103**
 MPLS
 BGP destinations 33
 configuration statements 39
 configuring 41
 CoS values 60
 fast reroute 38, 49, 87, 89
 IGP and BGP destinations 35
 link-layer protocols supported 20
 LSPs *See* LSPs
 overview 19
 per-prefix load balancing 38
 routing tables 36
 RSVP *See* RSVP
 signaled LSPs *See* signaled LSPs
 SNMP traps 82, 95
 standards supported 19

standby secondary paths 38
 static LSPs *See* static LSPs
 static MPLS 73, 87, 92, 106
 syslog messages 82, 95
 tracing protocol operations 83, 108
 traffic engineering overview 20
 traffic protection 38
 traffic statistics 81, 106
See also LDP; traffic engineering
 MPLS backbones, packet traversal 4, 24
 mpls statement **96**
 usage guidelines 11, 171
 MPLS tunneling
 CCC connection 175, 181
 CCC encapsulation 174
 example configurations 176
 overview 167, 173
 mpls.0 routing table 36, 76
 MTU sizes, MPLS tunneling 174
 mtu statement, usage guidelines 174
 Multiple Push (label operation) 23

N
 named paths
 empty paths 44, 100
 example configuration 45
 overview 44
 specifying routers 44
 names of LSPs 46
 neighbor (from operator, LDP) 148
 next hops, selecting 35
 nexthop (from operator, LDP) 148
 nexthop statement **96**
 usage guidelines 74, 76
 no-aggregate statement **129**
 no-cspf statement **97**
 usage guidelines 57
 no-decrement-ttl statement **97**
 usage guidelines 56
 non-point-to-point links 66
 no-propagate-ttl statement **98**
 usage guidelines 56
 no-record statement **103**
 usage guidelines 59
 normal (tracing flag) 109, 135
 no-stamp option 109, 135, 162, 315
 notification (tracing flag) 154, 161, 293
 notification messages, LDP 143
 no-world-readable option 109, 135, 162, 315

- offline path calculation 6, 27
 - operations on labels 23
 - optimize-aggressive statement **99**
 - usage guidelines 64
 - optimize-timer statement **99**
 - usage guidelines 63
 - optimizing LSPs 63, 99
 - oversubscription 124
- **P** packet forwarding component, traffic engineering 4
 - packet loss priority bit 60, 61
 - packet traversal on LSPs 4, 24
 - packet-dump (tracing flag) 154, 161, 162, 293
 - packets (tracing flag) 125, 134, 154, 161, 293
 - path (tracing flag) 125, 134, 154, 161, 293
 - path bandwidth, LSP 64
 - path calculation
 - constrained path computation 56, 97
 - CSPF algorithm 5, 25
 - offline path computation 6, 27
 - routing options 6
 - tie-breaking rules 27, 54, 103
 - path connection retry information 53, 104
 - path length, LSP 64, 91
 - Path messages, RSVP 117
 - path selection component, traffic engineering 5
 - path statement **100**
 - usage guidelines 44
 - PathErr messages, RSVP 118
 - pathtear (tracing flag) 125, 134
 - PathTear messages, RSVP 118
 - periodic (tracing flag) 154, 161, 293
 - per-prefix load balancing 38
 - PLP bit 60, 61
 - point-to-point links 66
 - policy (tracing flag) 109, 135
 - policy filters, LDP 140, 159
 - Pop (label operation) 23
 - pop statement **100**
 - usage guidelines 76
 - PPP circuits
 - Layer 2 switching cross-connects 170
 - MPLS tunneling cross-connects 175
 - preempting RSVP sessions 125
 - preemption, signaled LSPs 62, 102
 - preference levels
 - LDP routes 148, 161
 - signaled LSPs 59, 101
 - static LSPs 74, 76
 - preference statement
 - LDP **161**
 - usage guidelines 148
 - signaled LSPs **101**
 - usage guidelines 59
 - static LSPs **101**
 - usage guidelines 74, 76
 - prefix option 74
 - primary LSPs 49, 101
 - primary statement **101**
 - usage guidelines 49
 - priorities, signaled LSPs 62, 102
 - priority statement **102**
 - usage guidelines 62
 - Push (label operation) 23
 - push statement **102**
 - usage guidelines 74
- **Q** QoS requests 115
 - *See also* RSVP
- **R** random statement **103**
 - usage guidelines 54
 - random tie-breaking rule 27, 54, 103
 - receive (tracing flag modifier) 109, 135, 162, 315
 - received label filtering 140, 159
 - Record Route Object 59
 - record statement **103**
 - recording routes 59
 - refresh messages, RSVP 124
 - refresh time, RSVP 124
 - refresh-time statement **133**
 - usage guidelines 125
 - reject statement **104**
 - usage guidelines 76
 - remote-interface-switch statement **181**
 - usage guidelines 175
 - reoptimizing LSPs 63, 99
 - replace option 110, 136, 162, 315
 - requests, QoS 115
 - requests, QoS
 - *See also* RSVP
 - rerouting LSPs
 - adaptive rerouting 61, 85
 - fast reroute 38, 49, 87, 89
 - reserved labels 21
 - reserving network resources *See* RSVP
 - resource classes 52, 57
 - Resource Reservation Protocol *See* RSVP
 - resv (tracing flag) 125, 134
 - Resv messages, RSVP 118
 - ResvConfirm messages, RSVP 119

- ResvErr messages, RSVP 118
- resvtcar (tracing flag) 125, 134
- ResvTear messages, RSVP 118
- retry information 53, 104
- retry-limit statement **104**
 - usage guidelines 53
- retry-timer statement **104**
 - usage guidelines 53
- route (tracing flag) 109, 135
- route distinguishers, Layer 2 VPNs 208, 291, 313
- route preferences
 - LDP 148, 161
 - signaled LSPs 59, 101
- route-distinguisher statement **280, 313**
 - usage guidelines 208, 291
- Router Alert Label 21
- routers
 - egress routers 24, 75, 77, 92
 - ingress routers 24, 53, 68, 73, 87, 96, 106
 - label operations 23
 - LSP functions 24
 - transit routers 24
- routes
 - recording 59
 - route preferences 59, 101, 148, 161
- routing instances
 - configuring for VPNs 207
 - Layer 2 VPNs 289
- routing options, traffic engineering 6
- routing tables
 - IGP shortcuts 31
 - inet.0 31, 36
 - inet.3 31, 36, 52
 - installing host routes 52, 92
 - installing static routes 76
 - MPLS 36
 - mpls.0 36, 76
- routing-instances statement
 - usage guidelines 14
- RSVP
 - aggregation 122, 129
 - authentication 123, 130
 - bandwidth, reserving 124, 134
 - configuration statements 121, 129
 - configuring 122, 131, 133
 - disabling 122, 131
 - enabling 68, 122
 - enabling for VPNs 204
 - example configurations 126, 127
 - hello interval 123, 131
 - JUNOS implementation 116
 - Layer 2 VPNs 286
 - message types 117
 - MPLS, configuring with RSVP 127
 - overview 115
 - preemption 125
 - reservation styles 119
 - RFC draft documents 116
 - sessions 117, 127
 - signaled LSPs 24, 68
 - signalling extensions 6
 - timers 124, 133
 - tracing protocol traffic 125, 134
 - tunneling LDP LSPs through RSVP LSPs 95, 141, 152
 - See also LDP
 - rsvp statement **133**
 - usage guidelines 13, 122
- S**
 - scope of LSPs 25
 - SE (reservation style) 119
 - secondary LSPs 49, 64, 105
 - secondary paths 38
 - secondary statement **105**
 - usage guidelines 49
 - send (tracing flag modifier) 109, 135, 162, 315
 - session messages, LDP 142
 - sessions, RSVP 117, 127
 - setup priority, signaled LSPs 62
 - shared explicit reservation style 119
 - shared reservations 119
 - signaled LSPs
 - adaptive rerouting 61, 85
 - administrative groups 52, 57, 85, 86, 88, 91
 - associating addresses 52, 92
 - configuration statements 45, 94
 - constrained path computation 56, 97
 - CoS values 60
 - creating 45
 - damping LSP transitions 65
 - egress router address 48, 108
 - example configurations 68
 - fast reroute 49, 87, 89
 - fate sharing 66, 89
 - hold time 65, 86
 - ingress router address 48, 90
 - LDP tunneling 65
 - load balancing without CSPF 55
 - metrics 53, 54, 96
 - MPLS routers, configuring 68
 - named paths 44, 100
 - overview 43
 - path bandwidth 64
 - path connection retry information 53, 104
 - path length 64, 91
 - preemption 62, 102
 - preference levels 59, 101
 - primary LSPs 49, 101
 - priorities 62, 102
 - recording routes 59
 - reoptimization 63, 99
 - RSVP See RSVP
 - secondary LSPs 49, 105

standby state	64, 105
tie-breaking rules	27, 54, 103
TTL decrementing	55, 97, 98
signaling component, traffic engineering	6
signalling extensions, RSVP	6
site configuration, Layer 2 VPNs.....	314
site configuration, Layer 2 VPNs.....	292
site statement	314
usage guidelines.....	292
site-identifier statement	
usage guidelines.....	292
site-offset statement	
usage guidelines.....	292
size option.....	110, 136, 162, 315
SNMP traps, MPLS.....	82, 95
special labels	21
stacked labels	22
standby secondary paths.....	38
standby state, signaled LSPs.....	64, 105
standby statement	105
usage guidelines.....	64
state (tracing flag).....	109
state (tracing flag) 83, 109, 125, 134, 135, 154, 161, 293	
static LSPs	
configuring.....	73
egress routers.....	75, 77, 92
ingress routers	73, 87, 96, 106
intermediate routers	75, 92
overview	24
static LSP metric	54, 96
static MPLS.....	73, 87, 92, 106
static-path statement.....	106
usage guidelines.....	73
statistics output file	81
statistics statement	106
usage guidelines.....	81
statistics, MPLS traffic	81, 106
strict explicit routes.....	5, 79
strict option.....	44
subscribing to bandwidth	124, 134
subscription statement	134
usage guidelines.....	124
support, technical, contacting.....	xxvi
Swap (label operation).....	23
Swap and Push (label operation)	23
swap statement	107
usage guidelines.....	76
syslog messages, MPLS	82, 95

task (tracing flag)	109, 135
technical support, contacting.....	xxvi
tie-breaking rules, path calculation	27, 54, 103
timer (tracing flag)	109, 135
timers, RSVP	124, 133
to statement	108
usage guidelines.....	48
traceoptions statement	314
LDP	161
usage guidelines.....	154, 293
MPLS	108
usage guidelines.....	83
RSVP	134
usage guidelines.....	125
tracing protocol traffic	
LDP	154, 161, 293
MPLS	83, 108
RSVP	125, 134
traffic engineering	
BGP destinations	33
fate sharing.....	28
IGP and BGP destinations.....	35
IGP shortcuts.....	28
information distribution component	5
overview	3, 20
packet forwarding component	4
path selection component	5
routing options.....	6
signaling component.....	6
<i>See also</i> labels;MPLS	
traffic engineering database	26
traffic protection, MPLS	38
traffic statistics	81, 106
traffic-engineering statement	110
usage guidelines.....	56, 81
transit routers.....	24
transitions, damping	65
traps, SNMP.....	82, 95
TTL decrementing, disabling	55, 97, 98
tunneling through RSVP LSPs	95, 141, 152
tunneling, MPLS	
CCC encapsulation	174
example configurations.....	176
overview	167, 173
type statement.....	111
usage guidelines.....	76
typefaces, documentation conventions	xxix

U
undersubscription..... 124
unstable LSPs *See* fate sharing

V
Virtual Private Networks *See* VPNs
VPNs
 configuration statements..... 279–280, ??–316
 export policy 211, 280, 316
 IBGP session between PE routers 206
 import policy..... 210, 280, 316
 instance type..... 207, 279
 interface between PE and CD..... 279
 interface between PE and CE 207
 Layer 2 *See* Layer 2 VPNs
 Layer 3 *See* Layer 3 VPNs
 LDP, enabling..... 203
 route distinguisher 280
 routing instances..... 207
 RSVP, enabling..... 204
 signaling protocol, enabling..... 203
VPNs, implementing using MPLS..... 139
vrf-export statement **280, 316**
 usage guidelines 211
vrf-import statement **280, 316**
 usage guidelines 210

W
WF (reservation style)..... 119
wildcard filter reservation style..... 119
wildcard senders, RSVP 119
world-readable option 110, 136, 162, 315

Index

Index of Statements and Commands

A	
adaptive statement	85
admin-group statement	85
admin-groups statement.....	86
advertise-hold-time statement.....	86
aggregate statement.....	129
authentication-key statement.....	130
B	
bandwidth statement	
fast reroute.....	87
RSVP.....	130
signaled LSPs.....	87
C	
class-of-service statement	
ingress router configuration	87
signaled LSPs.....	87
static LSPs	87
connections statement	179
D	
disable statement	
LDP.....	157
MPLS	88
RSVP.....	131
discard statement	88
E	
encapsulation-type statement	311, 312
exclude statement	88
F	
fast-reroute statement.....	89
fate-sharing statement.....	89
from statement.....	90
H	
hello-interval statement	
LDP.....	158
RSVP.....	131
hold-time statement	159
hop-limit statement.....	91
I	
import statement	159
include statement	91
install statement	92
instance-type statement	279, 312
interface statement	279, 313
LDP.....	159
RSVP.....	131
static LSPs	92
interface-switch statement.....	180
K	
keepalive-interval statement	160
keepalive-timeout statement.....	160
keep-multiplier statement	132
L	
label-map statement	93
label-switched-path statement.....	94
ldp statement	160
ldp-tunneling statement.....	95
least-fill statement.....	103
log-updown statement.....	95
lsp-switch statement	180
M	
metric statement	96
most-fill statement	103
mpls statement.....	96

N		
next-hop statement.....	96	
no-aggregate statement.....	129	
no-cspf statement.....	97	
no-decrement-ttl statement.....	97	
no-propagate-ttl statement.....	98	
no-record statement.....	103	
O		
optimize-aggressive statement.....	99	
optimize-timer statement.....	99	
P		
path statement.....	100	
pop statement.....	100	
preference statement		
LDP.....	161	
static LSPs.....	101	
primary statement.....	101	
priority statement.....	102	
push statement.....	102	
R		
random statement.....	103	
record statement.....	103	
refresh-time statement.....	133	
reject statement.....	104	
remote-interface-switch statement.....	181	
retry-limit statement.....	104	
retry-timer statement.....	104	
route-distinguisher statement.....	280, 313	
rsvp statement.....	133	
S		
secondary statement.....	105	
site statement.....	314	
standby statement.....	105	
static-path statement.....	106	
statistics statement.....	106	
subscription statement.....	134	
swap statement.....	107	
T		
to statement.....	108	
traceoptions statement.....	314	
LDP.....	161	
MPLS.....	108	
RSVP.....	134	
traffic-engineering statement.....	110	
type statement.....	111	
V		
vrf-export statement.....	280, 316	
vrf-import statement.....	280, 316	