Junos® OS

Routing Protocols and Policies Configuration Guide for Security Devices

Release
12.1

Published: 2012-03-06

Copyright © 2012, Juniper Networks, Inc.

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

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 4.4BSD and 4.4BSD-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, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Junos OS Routing Protocols and Policies Configuration Guide for Security Devices
Release 12.1
Copyright © 2012, Juniper Networks, Inc.
All rights reserved.

Revision History
March 2012—R1 Junos OS 12.1

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS 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 website at www.juniper.net/techpubs.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Copyright © 2012, Juniper Networks, Inc.

Abbreviated Table of Contents

About This Guide xvii

Part 1 Routing Protocols

Chapter 1 Routing Overview 3
Chapter 2 Routing Instances 11
Chapter 3 Routing Tables 15
Chapter 4 Static Routing 23
Chapter 5 Aggregate and Generated Routes 91
Chapter 6 Packet Forwarding Behavior 105
Chapter 7 Martian Addresses 125
Chapter 8 RIP 131
Chapter 9 OSPF 159
Chapter 10 IS-IS 215
Chapter 11 BGP 227
Chapter 12 BFD 329
Chapter 13 Multicast 353

Part 2 Routing Policies and Stateless Firewall Filters

Chapter 14 Routing Policies 387
Chapter 15 Stateless Firewall Filters 419

Part 3 Debugging and Trace Operations

Chapter 16 Tracing Global Routing Protocol Operations 497

Part 4 Index

Index 505

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

Table of Contents

About This Guide . . . xvii

J Series and SRX Series Documentation and Release Notes . . . xvii
Objectives . . . xviii
Audience . . . xviii
Supported Routing Platforms . . . xviii
Document Conventions . . . xviii
Documentation Feedback . . . xx
Requesting Technical Support . . . xx
Self-Help Online Tools and Resources . . . xx
Opening a Case with JTAC . . . xxi

Part 1 Routing Protocols

Chapter 1 **Routing Overview . . . 3**

Routing Overview . . . 3
Networks and Subnetworks . . . 4
Autonomous Systems . . . 4
Interior and Exterior Gateway Protocols . . . 4
Routing Tables . . . 4
Forwarding Tables . . . 5
Dynamic and Static Routing . . . 6
Route Advertisements . . . 6
Route Aggregation . . . 7

Chapter 2 **Routing Instances . . . 11**

Routing Instances . . . 11
Routing Instances Overview . . . 11
Understanding Routing Instance Types . . . 12
Configuring Routing Instances . . . 13

Chapter 3 **Routing Tables . . . 15**

Routing Tables . . . 15
Understanding Junos OS Routing Tables . . . 15
Example: Creating Routing Tables . . . 16
Example: Importing Direct and Static Routes Into a Routing Instance . . . 18

Chapter 4 Static Routing 23
Basic Static Routes 23
Understanding Basic Static Routing 23
Example: Configuring a Basic Set of Static Routes 24
Example: Configuring IPv6 Static Routes 28
Static Routing on Logical Systems 32
Example: Creating an Interface on a Logical System 33
Example: Configuring Static Routes Between Logical Systems Within the Same Router 35
Example: Connecting Logical Systems Within the Same Router Using Logical Tunnel Interfaces 40
Example: Connecting a Logical System to a Physical Router 44
Static Route Selection 46
Understanding Static Route Preferences and Qualified Next Hops 46
Example: Configuring Static Route Preferences and Qualified Next Hops . . . 47
Static Route Control in Routing and Forwarding Tables 52
Understanding Static Route Control in Routing and Forwarding Tables 52
Route Retention 53
Readvertisement Prevention 53
Forced Rejection of Passive Route Traffic 53
Example: Preventing a Static Route from Being Readvertised 53
Static Routes and Point-to-Multipoint LSPs 59
Understanding Point-to-Multipoint LSPs 59
Point-to-Multipoint LSP Configuration Overview 60
Example: Configuring an RSVP-Signaled Point-to-Multipoint LSP 61
Static Routes and CLNs 79
Understanding Static Routes for CLNS 79
Example: Configuring Static Routes for CLNS 79
Default Static Route Behavior 82
Understanding Static Routing Default Properties 82
Example: Defining Default Behavior for All Static Routes 83
Verifying the Static Route Configuration 89
Chapter 5 Aggregate and Generated Routes 91
Understanding Aggregate Routes 91
Configuring a Metric Value for Aggregate Routes 94
Configuring a Preference Value for Aggregate Routes 94
Configuring the Next Hop for Aggregate Routes 95
Associating BGP Communities with Aggregate Routes 95
Associating AS Paths with Aggregate Routes 96
Including AS Numbers in Aggregate Route Paths 97
Configuring an OSPF Tag String for Aggregate Routes 97
Controlling Retention of Inactive Aggregate Routes in the Routing and Forwarding Tables 98
Example: Summarizing Routes Through Route Aggregation 98

 Copyright © 2012, Juniper Networks, Inc.

Chapter 6 Packet Forwarding Behavior 105
Indirect Next Hops and the Packet Forwarding Engine 105
Understanding Indirect Next Hops 105
Example: Optimizing Route Reconvergence by Enabling Indirect Next Hops on the Packet Forwarding Engine 106
Unicast Reverse-Path-Forwarding Check 115
Understanding Unicast Reverse Path Forwarding 115
Example: Configuring Unicast Reverse-Path-Forwarding Check 116
Chapter 7 Martian Addresses 125
Understanding Martian Addresses 125
Example: Configuring Martian Addresses 126
Chapter 8 RIP 131
RIP Overview 131
Distance-Vector Routing Protocols 131
Maximizing Hop Count 132
RIP Packets 133
Split Horizon and Poison Reverse Efficiency Techniques 133
Limitations of Unidirectional Connectivity 134
RIPng Overview 135
RIPng Protocol Overview 135
RIPng Standards 136
RIPng Packets 136
RIP Configuration Overview 137
Basic RIP Routing 137
Understanding Basic RIP Routing 137
Example: Configuring a Basic RIP Network 138
RIP Traffic Control Through Metrics 144
Understanding RIP Traffic Control with Metrics 144
Example: Controlling Traffic with an Incoming Metric 145
Example: Controlling Traffic with an Outgoing Metric 146
RIP Authentication 148
Understanding RIP Authentication 148
Enabling Authentication with Plain-Text Passwords (CLI Procedure) 149
Enabling Authentication with MD5 Authentication (CLI Procedure) 149
Verifying a RIP Configuration 150
Verifying the Exchange of RIP Messages 150
Verifying the RIP-Enabled Interfaces 151
Verifying Reachability of All Hosts in the RIP Network 152
RIP Demand Circuits Overview 153
RIP Demand Circuit Packets 154
Timers Used by RIP Demand Circuits 154
Example: Configuring RIP Demand Circuits 155
Chapter 9 OSPF 159
OSPF Overview 160
OSPF Default Route Preference Values 162
OSPF Routing Algorithm 162
OSPF Three-Way Handshake 163

Copyright © 2012, Juniper Networks, Inc.

OSPF Version 3 164
OSPF Configuration Overview 165
OSPF Designated Routers 166
OSPF Designated Router Overview 166
Example: Configuring an OSPF Router Identifier 167
Example: Controlling OSPF Designated Router Election 168
OSPF Areas, Area Border Routers, and Backbone Areas 170
Understanding OSPF Areas and Backbone Areas 170
Example: Configuring a Single-Area OSPF Network 172
Example: Configuring a Multiarea OSPF Network 174
OSPF Stub Areas and Not-So-Stubby Areas 176
Understanding OSPF Stub Areas, Totally Stubby Areas, and Not-So-Stubby Areas 176
Example: Configuring OSPF Stub and Totally Stubby Areas 178
Example: Configuring OSPF Not-So-Stubby Areas 182
OSPF Authentication 187
Understanding OSPFv2 Authentication 187
Example: Configuring Simple Authentication for OSPFv2 Exchanges 189
Example: Configuring MD5 Authentication for OSPFv2 Exchanges 191
Example: Configuring a Transition of MD5 Keys on an OSPFv2 Interface . . . 193
Example: Configuring IPsec Authentication for an OSPF Interface 196
OSPF Traffic Control 202
Understanding OSPF Traffic Control 202
Controlling the Cost of Individual OSPF Network Segments 202
Dynamically Adjusting OSPF Interface Metrics Based on Bandwidth . . 203
Controlling OSPF Route Preferences 203
Example: Controlling the Cost of Individual OSPF Network Segments 204
Example: Controlling OSPF Route Preferences 208
Verifying an OSPF Configuration 210
Verifying OSPF-Enabled Interfaces 210
Verifying OSPF Neighbors 211
Verifying the Number of OSPF Routes 211
Verifying Reachability of All Hosts in an OSPF Network 213

Chapter 10 IS-IS 215

IS-IS Overview 215
IS-IS Areas 215
System Identifier Mapping 216
ISO Network Addresses 216
IS-IS Path Selection 217
Protocol Data Units 217
IS-IS Hello PDU 217
Link-State PDU 218
Complete Sequence Number PDU 218
Partial Sequence Number PDU 218
Example: Configuring IS-IS 219
IS-IS Designated Routers 224
Understanding IS-IS Designated Routers 224
Configuring Designated Router Election Priority for IS-IS 224

 Copyright © 2012, Juniper Networks, Inc.

Chapter 11 BGP 227
Understanding BGP 228
Autonomous Systems 228
AS Paths and Attributes 228
External and Internal BGP 229
BGP Routes Overview 229
BGP Messages Overview 230
Open Messages 231
Update Messages 231
Keepalive Messages 232
Notification Messages 232
BGP Configuration Overview 232
BGP Local Preference 233
Understanding the BGP Local Preference 233
Example: Configuring the Local Preference Value for BGP Routes 233
BGP Peering Sessions 246
Understanding External BGP Peering Sessions 246
Example: Configuring External BGP Point-to-Point Peer Sessions 247
Example: Configuring Internal BGP Peer Sessions 254
BGP Path Selections 265
Understanding BGP Path Selection 265
Example: Advertising Multiple Paths in BGP 269
Understanding the Advertisement of Multiple Paths to a Single Destination in BGP 292
Example: Ignoring the AS Path Attribute When Selecting the Best Path . . . 293
BGP Multiple Exit Discriminator 300
Understanding the MED Attribute 301
Example: Always Comparing MEDs 303
BGP Route Reflectors 304
Understanding BGP Route Reflectors 304
Example: Configuring a Route Reflector 306
BGP Confederations 321
Understanding BGP Confederations 321
Example: Configuring BGP Confederations 322

Chapter 12 BFD 329
BFD and Static Routes 329
Understanding BFD for Static Routes 329
Example: Configuring BFD for Static Routes 332
Example: Enabling BFD on Qualified Next Hops in Static Routes 338
BFD Authentication and Static Routes 344
Understanding BFD Authentication for Static Routes 344
BFD Authentication Algorithms 344
Security Authentication Keychains 345
Strict Versus Loose Authentication 345
Example: Configuring BFD Authentication for Static Routes 346

Copyright © 2012, Juniper Networks, Inc.

Chapter 13 Multicast 353
Multicast Overview 353
Multicast Architecture 354
Upstream and Downstream Interfaces 354
Subnetwork Leaves and Branches 354
Multicast IP Address Ranges 355
Notation for Multicast Forwarding States 355
Dense and Sparse Routing Modes 355
Strategies for Preventing Routing Loops 356
Reverse-Path Forwarding for Loop Prevention 356
Shortest-Path Tree for Loop Prevention 356
Administrative Scoping for Loop Prevention 356
Multicast Protocol Building Blocks 357
Multicast Configuration Overview 359
SAP and SDP Multicast Session Announcements 360
Understanding SAP and SDP Multicast Session Announcements 360
Example: Configuring SAP and SDP to Listen for Session Announcements 361
Multicast IGMP 362
Understanding IGMP and Multicast 362
Example: Configuring IGMP for Multicast 362
IPv6 Multicast Flow 364
IPv6 Multicast Flow Overview 364
Multicast Listener Discovery (MLD) Overview 365
Multicast PIM and Static RPs 367
Understanding PIM and Static RPs 367
Example: Configuring PIM Sparse Mode and RP Static IP Addresses 368
PIM Register Messages 370
Understanding PIM Register Messages 370
Example: Rejecting Incoming PIM Register Messages on RP Routers 371
Example: Stopping Outgoing PIM Register Messages on a Designated Router 374
PIM RPF Routing Tables 377
Understanding PIM RPF Routing Tables 378
Example: Configuring a PIM RPF Routing Table 378
Verifying a Multicast Configuration 382
Verifying SAP and SDP Addresses and Ports 382
Verifying the IGMP Version 382
Verifying the PIM Mode and Interface Configuration 383
Verifying the PIM RP Configuration 383
Verifying the RPF Routing Table Configuration 384

 Copyright © 2012, Juniper Networks, Inc.

Part 2 Routing Policies and Stateless Firewall Filters

Chapter 14 Routing Policies 387

Routing Policies Overview 387
Routing Policies Configuration Overview 388
Routing Policies 389
Understanding Routing Policies 389
Example: Creating a Routing Policy 389
Routing Policy Terms 390
Understanding Routing Policy Terms 390
Example: Creating a Routing Policy Term 391
Routing Policy Match Conditions and Actions 392
Understanding Routing Policy Match Conditions and Actions 392
Match Conditions 392
Actions 394
Route-Based Match Conditions 396
Understanding Route-Based Match Conditions 396
Example: Rejecting Known Invalid Routes 397
Example: Grouping Source and Destination Prefixes into a Forwarding Class 399
Protocol-Based Match Conditions 401
Understanding Protocol-Based Match Conditions 402
Example: Injecting OSPF Routes into the BGP Routing Table 402
Autonomous System Path-Based Actions 405
Understanding Autonomous System Path-Based Actions 405
Example: Configuring a Routing Policy to Prepend the AS Path 406
Routing Policy Damping Parameters 408
Understanding Damping Parameters 408
Example: Configuring Damping Parameters 409
ECMP Flow-Based Forwarding 412
Understanding ECMP Flow-Based Forwarding 412
Example: Configuring ECMP Flow-Based Forwarding 413

Chapter 15 Stateless Firewall Filters 419

Introduction to Stateless Firewall Filters 419
Router Data Flow Overview 419
Flow of Routing Information 419
Flow of Data Packets 420
Flow of Local Packets 420
Interdependent Flows of Routing Information and Packets 420
Stateless Firewall Filter Overview 421
Packet Flow Control 421
Stateless and Stateful Firewall Filters 421
Purpose of Stateless Firewall Filters 422
Standard Stateless Firewall Filter Overview 422
Stateless Firewall Filter Types 423
Standard Stateless Firewall Filters 423
Service Filters 423

Copyright © 2012, Juniper Networks, Inc.

Simple Filters 423
Guidelines for Configuring Standard Firewall Filters 424
Statement Hierarchy for Configuring Standard Firewall Filters 424
Standard Firewall Filter Protocol Families 425
Standard Firewall Filter Names and Options 425
Standard Firewall Filter Terms 425
Standard Firewall Filter Match Conditions 426
Standard Firewall Filter Actions 427
Guidelines for Applying Standard Firewall Filters 429
Applying Standard Firewall Filters Overview 429
Applying a Firewall Filter to a Router's Physical Interfaces 429
Applying a Firewall Filter to the Router's Loopback Interface 429
Applying a Firewall Filter to Multiple Interfaces 429
Statement Hierarchy for Applying Standard Firewall Filters 429
Restrictions on Applying Standard Firewall Filters 430
Number of Input and Output Filters Per Logical Interface 430
MPLS and Layer 2 CCC Firewall Filters in Lists 430
Layer 2 CCC Firewall Filters on MX Series Routers 431
Protocol-Independent Firewall Filters on the Loopback Interface 431
Stateless Firewall Filter Terms 431
Stateless Firewall Filter Components 431
Protocol Family 431
Filter Type 432
Terms 433
Match Conditions 434
Actions 435
Standard Firewall Filter Match Conditions for IPv4 Traffic 436
Standard Firewall Filter Match Conditions for IPv6 Traffic 446
Firewall Filter Match Conditions Based on Bit-Field Values 453
Match Conditions for Bit-Field Values 453
Match Conditions for Common Bit-Field Values or Combinations 454
Logical Operators for Bit-Field Values 455
Matching on a Single Bit-Field Value or Text Alias 456
Matching on Multiple Bit-Field Values or Text Aliases 456
Matching on a Negated Bit-Field Value 456
Matching on the Logical OR of Two Bit-Field Values 457
Matching on the Logical AND of Two Bit-Field Values 457
Grouping Bit-Field Match Conditions 457
Firewall Filter Match Conditions Based on Numbers or Text Aliases 458
Matching on a Single Numeric Value 458
Matching on a Range of Numeric Values 458
Matching on a Text Alias for a Numeric Value 458
Matching on a List of Numeric Values or Text Aliases 458
Firewall Filter Match Conditions Based on Address Fields 459
Implied Match on the '0/0 except' Address for Firewall Filter Match Conditions Based on Address Fields 459
Matching an Address Field to a Subnet Mask or Prefix 459
Matching an Address Field to an Excluded Value 460
Matching Either IP Address Field to a Single Value 464

 Copyright © 2012, Juniper Networks, Inc.

Matching an Address Field to Noncontiguous Prefixes 464
Matching an Address Field to a Prefix List 466
Firewall Filter Match Conditions Based on Address Classes 467
Source-Class Usage 467
Destination-Class Usage 467
Guidelines for Applying SCU or DCU Firewall Filters to Output Interfaces 467
Standard Firewall Filter Terminating Actions 468
Standard Firewall Filter Nonterminating Actions 469
Trusted Source and Flood Prevention Stateless Firewall Filters 475
Understanding How to Use Standard Firewall Filters 475
Using Standard Firewall Filters to Affect Local Packets 475
Using Standard Firewall Filters to Affect Data Packets 476
Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources 476
Example: Configuring a Stateless Firewall Filter to Protect Against TCP and ICMP Floods 481
Fragment Handling Stateless Firewall Filters 488
Firewall Filters That Handle Fragmented Packets Overview 488
Example: Configuring a Stateless Firewall Filter to Handle Fragments 488
Example: Configuring a Filter to Match on IPv6 Flags 493

Part 3 Debugging and Trace Operations

Chapter 16 **Tracing Global Routing Protocol Operations 497**

Understanding Global Routing Protocol Tracing Operations 497
Example: Tracing Global Routing Protocol Operations 498

Part 4 Index

Index 505

Copyright © 2012, Juniper Networks, Inc.

About This Guide

This preface provides the following guidelines for using the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*:

- J Series and SRX Series Documentation and Release Notes on page xvii
- Objectives on page xviii
- Audience on page xviii
- Supported Routing Platforms on page xviii
- Document Conventions on page xviii
- Documentation Feedback on page xx
- Requesting Technical Support on page xx

J Series and SRX Series Documentation and Release Notes

For a list of related J Series documentation, see http://www.juniper.net/techpubs/software/junos-jseries/index-main.html .

For a list of related SRX Series documentation, see http://www.juniper.net/techpubs/hardware/srx-series-main.html .

If the information in the latest release notes differs from the information in the documentation, follow the *Junos OS Release Notes*.

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

Juniper Networks supports a technical book program to publish books by Juniper Networks engineers and subject matter experts with book publishers around the world. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration using the Junos operating system (Junos OS) and Juniper Networks devices. In addition, the Juniper Networks Technical Library, published in conjunction with O'Reilly Media, explores improving network security, reliability, and availability using Junos OS configuration techniques. All the books are for sale at technical bookstores and book outlets around the world. The current list can be viewed at http://www.juniper.net/books .

Objectives

This guide describes how to use and configure key security features on J Series Services Routers and SRX Series Services Gateways running Junos OS. It provides conceptual information, suggested workflows, and examples where applicable.

Audience

This manual is designed for anyone who installs, sets up, configures, monitors, or administers a J Series Services Router or an SRX Series Services Gateway running Junos OS. The manual is intended for the following audiences:

- Customers with technical knowledge of and experience with networks and network security, the Internet, and Internet routing protocols
- Network administrators who install, configure, and manage Internet routers

Supported Routing Platforms

This manual describes features supported on J Series Services Routers and SRX Series Services Gateways running Junos OS.

Document Conventions

Table 1 on page xviii defines the notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xix defines the text and syntax conventions used in this guide.

Copyright © 2012, Juniper Networks, Inc.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces important new terms. • Identifies book names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS System Basics Configuration Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; interface names; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose optional keywords or variables.	**stub** <**default-metric** *metric*>**;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** (***string1* | *string2* | *string3***)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

Indention and braces ({ })	Identify a level in the configuration hierarchy.	[edit]
		routing-options {
		static {
		route default {
		nexthop *address*;
		retain;
		}
		}
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

J-Web GUI Conventions

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to **techpubs-comments@juniper.net**, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need postsales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf .
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/ .
- JTAC Hours of Operation —The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Find product documentation: http://www.juniper.net/techpubs/

Copyright © 2012, Juniper Networks, Inc.

- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/ .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, visit us at http://www.juniper.net/support/requesting-support.html

Copyright © 2012, Juniper Networks, Inc.

PART 1

Routing Protocols

- Routing Overview on page 3
- Routing Instances on page 11
- Routing Tables on page 15
- Static Routing on page 23
- Aggregate and Generated Routes on page 91
- Packet Forwarding Behavior on page 105
- Martian Addresses on page 125
- RIP on page 131
- OSPF on page 159
- IS-IS on page 215
- BGP on page 227
- BFD on page 329
- Multicast on page 353

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 1

Routing Overview

- Routing Overview on page 3

Routing Overview

Routing is the transmission of data packets from a source to a destination address. It involves delivering a message across a network or networks. This process has two primary components: the exchange of routing information to forward packets accurately from source to destination and the packet-forwarding procedure.

For packets to be correctly forwarded to the appropriate host address, the host must have a unique numeric identifier or IP address. The unique IP address of the destination host forms entries in the routing table. These entries are primarily responsible for determining the path that a packet traverses when transmitted from source to destination.

To use the routing capabilities of a Juniper Networks device, you must understand the fundamentals of IP routing and the routing protocols that are primarily responsible for the transmission of unicast traffic. To understand this topic, you need a basic understanding of IP addressing and TCP/IP.

NOTE: **When configuring IPv6 addressing and routing on a J Series device, you must enable IPv6 in secure context .**

- **NetFlow V9 Support**

  NetFlow Services Export Version 9 (NetFlow V9) provides an extensible and flexible method for using templates to observe packets on a router. Each template indicates the format in which the router exports data.

  NetFlow V9 is supported in Junos OS in the adaptive service PIC module.

  This feature supports Netflow V5 or V8 for flow-based devices.

This topic contains the following sections:

- Networks and Subnetworks on page 4
- Autonomous Systems on page 4
- Interior and Exterior Gateway Protocols on page 4

Copyright © 2012, Juniper Networks, Inc.

- Routing Tables on page 4
- Forwarding Tables on page 5
- Dynamic and Static Routing on page 6
- Route Advertisements on page 6
- Route Aggregation on page 7

Networks and Subnetworks

Large groups of machines that are interconnected and can communicate with one another form networks. Typically, networks identify large systems of computers and devices that are owned or operated by a single entity. Traffic is routed between or through the networks as data is passed from host to host.

As networks grow large, the ability to maintain the network and effectively route traffic between hosts within the network becomes increasingly difficult. To accommodate growth, networks are divided into subnetworks. Fundamentally, subnetworks behave exactly like networks, except that they are identified by a more specific network address and subnet mask (destination prefix). Subnetworks have routing gateways and share routing information in exactly the same way as large networks.

Autonomous Systems

A large network or collection of routers under a single administrative authority is termed an *autonomous system* (AS). Autonomous systems are identified by a unique numeric identifier that is assigned by the Internet Assigned Numbers Authority (IANA). Typically, the hosts within an AS are treated as internal peers, and hosts in a peer AS are treated as external peers. The status of the relationship between hosts—internal or external—governs the protocol used to exchange routing information.

Interior and Exterior Gateway Protocols

Routing information that is shared within an AS is transmitted by an interior gateway protocol (IGP). Of the different IGPs, the most common are RIP, OSPF, and IS-IS. IGPs are designed to be fast acting and light duty. They typically incorporate only a moderate security system, because trusted internal peers do not require the stringent security measures that untrusted peers require. As a result, you can usually begin routing within an AS by enabling the IGP on all internal interfaces and performing minimal additional configuration. You do not need to establish individual adjacencies.

Routing information that is shared with a peer AS is transmitted by an exterior gateway protocol (EGP). The primary EGP in use in almost all networks is the Border Gateway Protocol (BGP). BGP is designed to be very secure. Individual connections must be explicitly configured on each side of the link. As a result, although large numbers of connections are difficult to configure and maintain, each connection is secure.

Routing Tables

To route traffic from a source host to a destination host, the devices through which the traffic will pass must learn the path that the packet is to take. Once learned, the information is stored in routing tables. The routing table maintains a list of all the possible paths from point A to point B. Figure 1 on page 5 shows a simple network of routers.

 Copyright © 2012, Juniper Networks, Inc.

Figure 1: Simple Network Topology

This simple network provides multiple ways to get from host San Francisco to host Miami. The packet can follow the path through Denver and Cleveland. Alternatively, the packet can be routed through Phoenix and directly to Miami. The routing table includes all the possible paths and combinations—an exhaustive list of all the ways to get from the source to the destination.

The routing table must include every possible path from a source to a destination. Routing tables for the network in Figure 1 on page 5 must include entries for San Francisco-Denver, San Francisco-Cleveland, San Francisco-Miami, Denver-Cleveland, and so on. As the number of sources and destinations increases, the routing table quickly becomes large. The unwieldy size of routing tables is the primary reason for the division of networks into subnetworks.

Forwarding Tables

If the routing table is a list of all the possible paths a packet can take, the forwarding table is a list of only the best routes to a particular destination. The best path is determined according to the particular routing protocol being used, but generally the number of hops between the source and destination determines the best possible route.

In the network shown in Figure 1 on page 5, because the path with the fewest number of hops from San Francisco to Miami is through Phoenix, the forwarding table distills all the possible San Francisco-Miami routes into the single route through Phoenix. All traffic with a destination address of Miami is sent directly to the next hop, Phoenix.

After it receives a packet, the Phoenix router performs another route lookup, using the same destination address. The Phoenix router then routes the packet appropriately. Although it considers the entire path, the router at any individual hop along the way is responsible only for transmitting the packet to the next hop in the path. If the Phoenix router is managing its traffic in a particular way, it might send the packet through Houston on its route to Miami. This scenario is likely if specific customer traffic is treated as priority traffic and routed through a faster or more direct route, while all other traffic is treated as nonpriority traffic.

Copyright © 2012, Juniper Networks, Inc.

Dynamic and Static Routing

Entries are imported into a router's routing table from dynamic routing protocols or by manual inclusion as static routes. Dynamic routing protocols allow routers to learn the network topology from the network. The routers within the network send out routing information in the form of route advertisements. These advertisements establish and communicate active destinations, which are then shared with other routers in the network.

Although dynamic routing protocols are extremely useful, they have associated costs. Because they use the network to advertise routes, dynamic routing protocols consume bandwidth. Additionally, because they rely on the transmission and receipt of route advertisements to build a routing table, dynamic routing protocols create a delay (latency) between the time a router is powered on and the time during which routes are imported into the routing table. Some routes are therefore effectively unavailable until the routing table is completely updated, when the router first comes online or when routes change within the network (due to a host going offline, for example).

Static routing avoids the bandwidth cost and route import latency of dynamic routing. Static routes are manually included in the routing table, and never change unless you explicitly update them. Static routes are automatically imported into the routing table when a router first comes online. Additionally, all traffic destined for a static address is routed through the same router. This feature is particularly useful for networks with customers whose traffic must always flow through the same routers. Figure 2 on page 6 shows a network that uses static routes.

Figure 2: Static Routing Example

In Figure 2 on page 6, the customer routes in the **192.176.14/24** subnetwork are static routes. These are hard links to specific customer hosts that never change. Because all traffic destined for any of these routes is forwarded through Router A, these routes are included as static routes in Router A's routing table. Router A then advertises these routes to other hosts so that traffic can be routed to and from them.

Route Advertisements

The routing table and forwarding table contain the routes for the routers within a network. These routes are learned through the exchange of route advertisements. Route advertisements are exchanged according to the particular protocol being employed within the network.

Generally, a router transmits hello packets out each of its interfaces. Neighboring routers detect these packets and establish adjacencies with the router. The adjacencies are then shared with other neighboring routers, which allows the routers to build up the entire network topology in a topology database, as shown in Figure 3 on page 7.

Copyright © 2012, Juniper Networks, Inc.

Figure 3: Route Advertisement

In Figure 3 on page 7, Router A sends out hello packets to each of its neighbors. Routers B and C detect these packets and establish an adjacent relationship with Router A. Router B and C then share this information with their neighbors, Routers D and E, respectively. By sharing information throughout the network, the routers create a network topology, which they use to determine the paths to all possible destinations within the network. The routes are then distilled into the forwarding table of best routes according to the route selection criteria of the protocol in use.

Route Aggregation

As the number of hosts in a network increases, the routing and forwarding tables must establish and maintain more routes. As these tables become larger, the time routers require to look up particular routes so that packets can be forwarded becomes prohibitive. The solution to the problem of growing routing tables is to group (aggregate) the routers by subnetwork, as shown in Figure 4 on page 8.

Copyright © 2012, Juniper Networks, Inc.

Figure 4: Route Aggregation

Figure 4 on page 8 shows three different ASs. Each AS contains multiple subnetworks with thousands of host addresses. To allow traffic to be sent from any host to any host, the routing tables for each host must include a route for each destination. For the routing tables to include every combination of hosts, the flooding of route advertisements for each possible route becomes prohibitive. In a network of hosts numbering in the thousands or even millions, simple route advertisement is not only impractical but impossible.

By employing route aggregation, instead of advertising a route for each host in AS 3, the gateway router advertises only a single route that includes all the routes to all the hosts within the AS. For example, instead of advertising the particular route **170.16.124.17**, the AS 3 gateway router advertises only **170.16/16**. This single route advertisement encompasses all the hosts within the **170.16/16** subnetwork, which reduces the number of routes in the routing table from 2^{16} (one for every possible IP address within the subnetwork) to 1. Any traffic destined for a host within the AS is forwarded to the gateway router, which is then responsible for forwarding the packet to the appropriate host.

Similarly, in this example, the gateway router is responsible for maintaining 2^{16} routes within the AS (in addition to any external routes). The division of this AS into subnetworks allows for further route aggregation to reduce this number. In the subnetwork in the example, the subnetwork gateway router advertises only a single route (**170.16.124/24**), which reduces the number of routes from 2^{8} to 1.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Instances Overview on page 11
- Multicast Overview on page 353
- Routing Policies Overview on page 387
- IPv6 Overview in the *Junos OS Routing Protocols Configuration Guide*

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 2

Routing Instances

- Routing Instances on page 11

Routing Instances

- Routing Instances Overview on page 11
- Understanding Routing Instance Types on page 12
- Configuring Routing Instances on page 13

Routing Instances Overview

A *routing instance* is a collection of routing tables, interfaces, and routing protocol parameters. There can be multiple routing tables for a single routing instance—for example, unicast IPv4, unicast IPv6, and multicast IPv4 routing tables can exist in a single routing instance. Routing protocol parameters and options control the information in the routing tables.

The default routing instance, master, refers to the main inet.0 routing table. The master routing instance is reserved and cannot be specified as a routing instance. Routes are installed into the master routing instance by default, unless a routing instance is specified. Configure global routing options and protocols for the master routing instance by including statements at the **[edit protocols]** and **[edit routing-options]** hierarchy levels.

You can configure six types of routing instances on SRX Series and J Series devices: forwarding, Layer 2 virtual private network (VPN), nonforwarding, VPN routing and forwarding (VRF), virtual router, and virtual private LAN service (VPLS).

Each routing instance has a unique name and a corresponding IP unicast routing table. For example, if you create a routing instance with the name my-instance, the corresponding IPv4 unicast routing table is my-instance.inet.0. All IPv4 routes for my-instance are installed into my-instance.inet.0.

Each routing instance consists of sets of the following:

- Routing tables
- Interfaces that belong to these routing tables (optional, depending upon the routing instance type)
- Routing option configurations

Copyright © 2012, Juniper Networks, Inc.

You can create multiple instances of BGP, IS-IS, Multicast Source Discovery Protocol (MSDP), OSPF version 2 (usually referred to simply as OSPF), OSPF version 3 (OSPFv3), Protocol Independent Multicast (PIM), Routing Information Protocol (RIP), RIP next generation (RIPng), and static routes in a device by including statements at the **[edit routing-instances *routing-instance-name* protocols]** hierarchy level. Only one instance of each protocol can be configured in single routing instance.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Overview on page 3
- Understanding Routing Instance Types on page 12
- Configuring Routing Instances on page 13

Understanding Routing Instance Types

You can configure six types of routing instances on SRX Series and J Series devices:

- Forwarding—Use for filter-based forwarding applications. For this instance type, there is no one-to-one mapping between an interface and a routing instance. All interfaces belong to the default instance inet.0. See the *Junos OS Routing Protocols Configuration Guide*.
- Layer 2 virtual private network (VPN)—Use for Layer 2 virtual private network (VPN) implementations. See the *Junos OS MPLS Configuration Guide for Security Devices*.
- Nonforwarding—Use when a separation of routing table information is required. There is no corresponding forwarding table. All routes are installed into the default forwarding table. IS-IS instances are strictly nonforwarding instance types.

  Nonforwarding instances of IS-IS and OSPF can be used to separate a very large network into smaller administrative entities. Instead of configuring a large number of filters, nonforwarding instances can be used to filter routes, thereby instantiating policies. Nonforwarding instances can be used to reduce the amount of routing information advertised throughout all components of a network. Routing information associated with a particular instance can be announced where required, instead of being advertised to the whole network. See the *Junos OS Routing Protocols Configuration Guide*.
- VPN routing and forwarding (VRF)—Use for Layer 3 VPN implementations. The VRF routing instance type has a VPN routing table as well as a corresponding VPN forwarding table. For this instance type, there is a one-to-one mapping between an interface and a routing instance. Routes on an interface go into the corresponding forwarding table. (VRF is sometimes known as a virtual router and forwarding instance.)

  Multiple instances of BGP, OSPF, and RIP are used for Layer 3 VPN implementation. Routing information for different VPNs are kept separate. The VRF instance advertises routes from the customer edge (CE) router to the provider edge (PE) router and advertises routes from the PE router to the CE router. Each VPN receives only routing information belonging to that VPN. See the *Junos OS MPLS Configuration Guide for Security Devices*.

 Copyright © 2012, Juniper Networks, Inc.

- Virtual router—Use for non-VPN related applications. It is similar to a VRF instance type. There are no VRF import, VRF export, VRF target, or route distinguisher requirements for this instance type. See the *Junos OS Security Configuration Guide*.
- Virtual private LAN service (VPLS)—Use for point-to-multipoint LAN implementations between a set of sites in a VPN. See the *Junos OS MPLS Configuration Guide for Security Devices*.

To configure a routing instance type, use the **instance-type** statement at the [**edit routing-instances** ***routing-instance-name***] hierarchy level.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Overview on page 3
- Routing Instances Overview on page 11
- Configuring Routing Instances on page 13

Configuring Routing Instances

To configure a routing instance, specify the following parameters:

- Name of the routing instance. Each routing instance has a unique name and a corresponding IP unicast table. For example, if you configure a routing instance with the name my-instance, its corresponding IP unicast table is my-instance.inet.0. All routes for my-instance are installed into **my-instance.inet.0**.

  NOTE: **You cannot specify a routing-instance name of *default* or include special characters within the name of a routing instance.**

- Type of routing instance.
- The interfaces that are bound to the routing instance. Interfaces not required for the forwarding routing instance type.

To configure a routing instance, use the **routing-instances** statement at the [**edit**] hierarchy level.

You can create an instance of BGP, IS-IS, OSPF, OSPFv3, RIP, or RIPng by including configuration statements at the **[edit routing-instances *routing-instance-name* protocols]** hierarchy level. You can also configure static routes for the routing instance.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Overview on page 3
- Routing Instances Overview on page 11
- Understanding Routing Instance Types on page 12
- *Junos OS Interfaces Configuration Guide for Security Devices*

Copyright © 2012, Juniper Networks, Inc.

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 3

Routing Tables

- Routing Tables on page 15

Routing Tables

- Understanding Junos OS Routing Tables on page 15
- Example: Creating Routing Tables on page 16
- Example: Importing Direct and Static Routes Into a Routing Instance on page 18

Understanding Junos OS Routing Tables

Junos OS automatically creates and maintains several routing tables. Each routing table is used for a specific purpose. In addition to these automatically created routing tables, you can create your own routing tables.

Each routing table populates a portion of the forwarding table. Thus, the forwarding table is partitioned based on routing tables. This allows for specific forwarding behavior for each routing table. For example, for VPNs, each VPN-based routing table has its own VPN-specific partition in the forwarding table.

It is common for the routing software to maintain unicast routes and multicast routes in different routing tables. You also might have policy considerations that would lead you to create separate routing tables to manage the propagation of routing information.

Creating routing tables is optional. If you do not create any, Junos OS uses its default routing tables, which are as follows:

- **inet.0**—For IP version 4 (IPv4) unicast routes. This table stores interface local and direct routes, static routes, and dynamically learned routes.
- **inet.1**—For the IPv4 multicast forwarding cache. This table stores the IPv4 (S,G) group entries that are dynamically created as a result of join state information.
- **inet.2**—For subsequent address family indicator (SAFI) 2 routes, when multiprotocol BGP (MBGP) is enabled. This table stores unicast routes that are used for multicast reverse-path-forwarding (RPF) lookup. The routes in this table can be used by the Distance Vector Multicast Routing Protocol (DVMRP), which requires a specific RPF table. In contrast, Protocol Independent Multicast (PIM) does not need this table because it can perform RPF checks against the inet.0 table. You can import routes

Copyright © 2012, Juniper Networks, Inc.

from inet.0 into inet.2 using routing information base (RIB) groups, or install routes directly into inet.2 from a multicast routing protocol.

- **inet.3**—For IPv4 MPLS. This table stores the egress address of an MPLS label-swiched path (LSP), the LSP name, and the outgoing interface name. This routing table is used only when the local device is the ingress node to an LSP.
- **inet6.0**—For IP version 6 (IPv6) unicast routes. This table stores interface local and direct routes, static routes, and dynamically learned routes.
- ***instance-name*.inet.0**—If you configure a routing instance, Junos OS creates the default unicast routing table ***instance-name*.inet.0**.
- ***instance-name*.inetflow.0**—If you configure a flow route, Junos OS creates the flow routing table ***instance-name*.inetflow.0**.
- **bgp.l2vpn.0**—For Layer 2 VPN routes learned from BGP. This table stores routes learned from other provider edge (PE) routers. The Layer 2 routing information is copied into Layer 2 VPN routing and forwarding instances (VRFs) based on target communities.
- **bgp.l3vpn.0**—For Layer 3 VPN routes learned from BGP. This table stores routes learned from other PE routers. Routes in this table are copied into a Layer 3 VRF when there is a matching route table.
- **mpls.0**—For MPLS label switching operations. This table is used when the local device is a transit router.
- **iso.0**—For IS-IS routes. When you are using IS-IS to support IP routing, this table contains only the local device's network entity title (NET).
- **juniper_private**—For Junos OS to communicate internally between the Routing Engine and PIC hardware.

Related Documentation

- Example: Creating Routing Tables on page 16

Example: Creating Routing Tables

This example shows how to create a custom routing table.

- Requirements on page 16
- Overview on page 16
- Configuration on page 17
- Verification on page 18

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

Creating routing tables is optional. You might have policy considerations that would lead you to create separate routing tables to manage the propagation of routing information. This capability is rarely used, but it is demonstrated here for completeness.

 Copyright © 2012, Juniper Networks, Inc.

If you do not create any routing tables, Junos OS uses its default routing tables.

NOTE: If you want to add static, aggregate, generated, or martian routes only to the default IPv4 unicast routing table (inet.0), you do not have to create any routing tables because, by default, these routes are added to inet.0. You can add these routes by including the static, aggregate, generate, and martians statements.

To explicitly create a routing table, include the **rib** statement and child statements under the **rib** statement.

The routing table name, ***routing-table-name***, includes the protocol family, optionally followed by a period and a number. The protocol family can be **inet** for the IPv4 family, **inet6** for the IPv6 family, or **iso** for the International Standards Organization (ISO) protocol family. The number represents the routing instance. The first instance is 0.

This example shows how to configure a custom IPv4 routing table called inet.14. The example also shows how to populate the routing table with a single static route.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set routing-options rib inet.14 static route 10.2.0.0/16 discard
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To create a routing table:

1. Configure the routing table.

```
[edit routing-options]
user@host# set rib inet.14 static route 10.2.0.0/16 discard
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

Confirm your configuration by issuing the **show routing-options** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show routing-options
rib inet.14 {
   static {
      route 10.2.0.0/16 discard;
```

```
    }
  }
```

Verification

Confirm that the configuration is working properly.

Checking the Routing Table

Purpose Make sure that the static route appears in the custom routing table.

Action

```
user@host> show route table inet.14
inet.14: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.2.0.0/16        *[Static/5] 00:00:09
                      Discard
```

Meaning The static route is in the custom routing table.

Related Documentation

- Understanding Junos OS Routing Tables on page 15

Example: Importing Direct and Static Routes Into a Routing Instance

This example shows how to populate the routing table that is created when you configure a virtual router.

- Requirements on page 18
- Overview on page 18
- Configuration on page 19
- Verification on page 21

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

You can install routes into more than one routing table. For example, you might want a simple configuration that allows you to install a static route into the default routing table **inet.0**, as well as a second routing table **vpna.inet.0**. Instead of configuring the same static route for each routing table, you can use routing table groups to insert the route into multiple tables. To create a routing table group, include the **rib-groups** statement.

This example shows how to export static routes, direct routes, and local routes from the default IPv4 unicast routing table (**inet.0**) and import them into the IPv4 unicast routing table of a virtual router called vpna (**vpna.inet.0**).

 Copyright © 2012, Juniper Networks, Inc.

NOTE: To explicitly create a routing table, include the **rib** statement. In this case, you do not need to use the **rib** statement because when you configure a routing instance, Junos OS automatically creates the routing table *instance-name*.inet.0.

In this example, Device A and Device B are directly connected to each other. Device A also has a virtual router configured called vpna. Device A's **inet.0** routing table has direct and local routes (also known as interface routes). These routes are imported into vpna's **inet.0** routing table (**vpna.inet.0**). Device A also has a static route configured. This static route is also imported into **vpna.inet.0**.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device A

```
set interfaces fe-1/2/0 unit 4 description A->B
set interfaces fe-1/2/0 unit 4 family inet address 10.0.2.2/30
set interfaces lo0 unit 1 family inet address 1.1.1.1/32
set routing-instances vpna instance-type virtual-router
set routing-options interface-routes rib-group inet group1
set routing-options static rib-group group1
set routing-options static route 192.168.1.0/24 discard
set routing-options rib-groups group1 import-rib inet.0
set routing-options rib-groups group1 import-rib vpna.inet.0
```

Device B

```
set interfaces fe-1/2/0 unit 3 description B->A
set interfaces fe-1/2/0 unit 3 family inet address 10.0.2.1/30
set interfaces lo0 unit 2 family inet address 2.2.2.2/32
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure this example:

1. Configure the routing instance.

   ```
   [edit routing-instances]
   user@A# set routing-instances vpna instance-type virtual-router
   ```

2. Configure the interfaces.

   ```
   [edit interfaces]
   user@A# set interfaces fe-1/2/0 unit 4 description A->B
   user@A# set interfaces fe-1/2/0 unit 4 family inet address 10.0.2.2/30
   user@A# set interfaces lo0 unit 1 family inet address 1.1.1.1/32
   ```

3. Configure one or more static routes.

   ```
   [edit routing-options]
   user@A# set routing-options static route 192.168.1.0/24 discard
   ```

Copyright © 2012, Juniper Networks, Inc.

4. Include the direct and local routes in a routing table group called group1.

   The **interface-routes** statement specifies the direct and local routes to match against.

   ```
   [edit routing-options]
   user@A# set routing-options interface-routes rib-group inet group1
   ```

5. Include all static routes in the group1 routing table group.

   The **static** statement specifies the protocol (static) to match against.

   ```
   [edit routing-options]
   user@A# set routing-options static rib-group group1
   ```

6. Configure the primary routing table for group1.

   The primary routing table determines the address family of the routing table group. To configure an IPv4 group, specify **inet.0** as the primary table. To configure an IPv6 group, specify **inet6.0** as the primary routing table.

   ```
   [edit routing-options]
   user@A# set routing-options rib-groups group1 import-rib inet.0
   ```

7. Configure the secondary routing table for group1.

   ```
   [edit routing-options]
   user@A# set routing-options rib-groups group1 import-rib vpna.inet.0
   ```

8. If you are done configuring the device, commit the configuration.

   ```
   [edit routing-options]
   user@A# commit
   ```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device B:

1. Configure the interfaces.

   ```
   [edit interfaces]
   user@B# set fe-1/2/0 unit 3 description B->A
   user@B# set fe-1/2/0 unit 3 family inet address 10.0.2.1/30

   user@B# set lo0 unit 2 family inet address 2.2.2.2/32
   ```

2. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@B# commit
   ```

Results

Confirm your configuration by issuing the **show interfaces**, **show routing-instances**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@A# show interfaces
fe-1/2/0 {
   unit 4 {
```


Copyright © 2012, Juniper Networks, Inc.

```
        description A->B;
        family inet {
          address 10.0.2.2/30;
        }
      }
    }
    lo0 {
      unit 1 {
        family inet {
          address 1.1.1.1/32;
        }
      }
    }

    user@A# show routing-instances
    vpna {
      instance-type virtual-router;
    }

    user@A# show routing-options
    interface-routes {
      rib-group inet group1;
    }
    static {
      rib-group group1;
      route 192.168.1.0/24 discard;
    }
    rib-groups {
      group1 {
        import-rib [ inet.0 vpna.inet.0 ];
      }
    }

    user@B# show interfaces
    fe-1/2/0 {
      unit 3 {
        description B->A;
        family inet {
          address 10.0.2.1/30;
        }
      }
    }
    lo0 {
      unit 2 {
        family inet {
          address 2.2.2.2/32;
        }
      }
    }
```

Verification

Confirm that the configuration is working properly.

Checking the Routing Tables

Purpose Make sure that the expected routes appear in the routing tables.

Action

```
user@A> show route table inet.0

inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1/32         *[Direct/0] 02:51:24
                    > via lo0.1
10.0.2.0/30        *[Direct/0] 03:19:06
                    > via fe-1/2/0.4
10.0.2.2/32        *[Local/0] 03:19:07
                       Local via fe-1/2/0.4
192.168.1.0/24     *[Static/5] 00:44:21
                       Discard

user@A> show route table vpna.inet.0

vpna.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1/32         *[Direct/0] 02:35:39
                    > via lo0.1
10.0.2.0/30        *[Direct/0] 02:35:39
                    > via fe-1/2/0.4
10.0.2.2/32        *[Local/0] 02:35:39
                       Local via fe-1/2/0.4
192.168.1.0/24     *[Static/5] 00:44:28
                       Discard
```

Meaning

The static route and the interface routes appear in both routing tables.

Related Documentation

- Understanding Junos OS Routing Tables on page 15

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 4

Static Routing

- Basic Static Routes on page 23
- Static Routing on Logical Systems on page 32
- Static Route Selection on page 46
- Static Route Control in Routing and Forwarding Tables on page 52
- Static Routes and Point-to-Multipoint LSPs on page 59
- Static Routes and CLNs on page 79
- Default Static Route Behavior on page 82
- Verifying the Static Route Configuration on page 89

Basic Static Routes

- Understanding Basic Static Routing on page 23
- Example: Configuring a Basic Set of Static Routes on page 24
- Example: Configuring IPv6 Static Routes on page 28

Understanding Basic Static Routing

Routes that are permanent fixtures in the routing and forwarding tables are often configured as static routes. These routes generally do not change, and often include only one or very few paths to the destination.

To create a static route in the routing table, you must, at minimum, define the route as static and associate a next-hop address with it. The static route in the routing table is inserted into the forwarding table when the next-hop address is reachable. All traffic destined for the static route is transmitted to the next-hop address for transit.

You can specify options that define additional information about static routes that is included with the route when it is installed in the routing table. All static options are optional.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring a Basic Set of Static Routes on page 24

Example: Configuring a Basic Set of Static Routes

This example shows how to configure a basic set of static routes.

- Requirements on page 24
- Overview on page 24
- Configuration on page 25
- Verification on page 27

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

There are many practical applications for static routes. Static routing is often used at the network edge to support attachment to stub networks, which, given their single point of entry and egress, are well suited to the simplicity of a static route. In Junos OS, static routes have a global preference of 5. Static routes are activated if the specified next hop is reachable.

In this example, you configure the static route 192.168.47.0/24 from the provider network to the customer network, using the next-hop address of 172.16.1.2. You also configure a static default route of 0.0.0.0/0 from the customer network to the provider network, using a next-hop address of 172.16.1.1.

For demonstration purposes, some loopback interfaces are configured on Device B and Device D. These loopback interfaces provide addresses to ping and thus verify that the static routes are working.

Figure 5 on page 25 shows the sample network.

Copyright © 2012, Juniper Networks, Inc.

Figure 5: Customer Routes Connected to a Service Provider

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device B

```
set interfaces ge-1/2/0 unit 0 description B->D
set interfaces ge-1/2/0 unit 0 family inet address 172.16.1.1/24
set interfaces lo0 unit 57 family inet address 10.0.0.1/32
set interfaces lo0 unit 57 family inet address 10.0.0.2/32
set routing-options static route 192.168.47.0/24 next-hop 172.16.1.2
```

Device D

```
set interfaces ge-1/2/0 unit 1 description D->B
set interfaces ge-1/2/0 unit 1 family inet address 172.16.1.2/24
set interfaces lo0 unit 2 family inet address 192.168.47.5/32
set interfaces lo0 unit 2 family inet address 192.168.47.6/32
set routing-options static route 0.0.0.0/0 next-hop 172.16.1.1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure basic static routes:

1. On Device B, configure the interfaces.

```
[edit interfaces]
user@B# set ge-1/2/0 unit 0 description B->D
user@B# set ge-1/2/0 unit 0 family inet address 172.16.1.1/24
user@B# set lo0 unit 57 family inet address 10.0.0.1/32
user@B# set lo0 unit 57 family inet address 10.0.0.2/32
```

2. On Device B, create a static route and set the next-hop address.

```
[edit routing-options]
user@B# set static route 192.168.47.0/24 next-hop 172.16.1.2
```

3. If you are done configuring Device B, commit the configuration.

```
[edit interfaces]
user@B# commit
```

4. On Device D, configure the interfaces.

```
[edit]
user@D# set ge-1/2/0 unit 1 description D->B
user@D# set ge-1/2/0 unit 1 family inet address 172.16.1.2/24
user@D# set lo0 unit 2 family inet address 192.168.47.5/32
user@D# set lo0 unit 2 family inet address 192.168.47.6/32
```

5. On Device D, create a static route and set the next-hop address.

```
[edit routing-options]
user@D# set static route 0.0.0.0/0 next-hop 172.16.1.1
```

6. If you are done configuring Device D, commit the configuration.

```
[edit]
user@D# commit
```

Results Confirm your configuration by issuing the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device B

```
user@B# show interfaces
ge-1/2/0 {
   unit 0 {
      description B->D;
      family inet {
         address 172.16.1.1/24;
      }
   }
}
lo0 {
   unit 57 {
      family inet {
         address 10.0.0.1/32;
         address 10.0.0.2/32;
      }
   }
}

user@B# show routing-options
static {
   route 192.168.47.0/24 next-hop 172.16.1.2;
}
```

Device D

```
user@D# show interfaces
ge-1/2/0 {
   unit 1 {
      description D->B;
```

 Copyright © 2012, Juniper Networks, Inc.

```
        family inet {
          address 172.16.1.2/24;
        }
      }
    }
    lo0 {
      unit 2 {
        family inet {
          address 192.168.47.5/32;
          address 192.168.47.6/32;
        }
      }
    }

    user@D# show routing-options
    static {
      route 0.0.0.0/0 next-hop 172.16.1.1;
    }
```

Verification

Confirm that the configuration is working properly.

- Checking the Routing Tables on page 27
- Pinging the Remote Addresses on page 28

Checking the Routing Tables

Purpose Make sure that the static routes appear in the routing tables of Device B and Device D.

Action

```
user@B> show route
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.1/32        *[Direct/0] 00:29:43
                    > via lo0.57
10.0.0.2/32        *[Direct/0] 00:29:43
                    > via lo0.57
172.16.1.0/24      *[Direct/0] 00:34:40
                    > via ge-1/2/0.0
172.16.1.1/32      *[Local/0] 00:34:40
                       Local via ge-1/2/0.0
192.168.47.0/24   *[Static/5] 00:31:23
         > to 172.16.1.2 via ge-1/2/0.0

user@D> show route
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0       *[Static/5] 00:31:24
         > to 172.16.1.1 via ge-1/2/0.1
172.16.1.0/24      *[Direct/0] 00:35:21
                    > via ge-1/2/0.1
172.16.1.2/32      *[Local/0] 00:35:21
                       Local via ge-1/2/0.1
192.168.47.5/32    *[Direct/0] 00:35:22
                    > via lo0.2
```

```
192.168.47.6/32    *[Direct/0] 00:35:21
                    > via lo0.2
```

Meaning The static routes are in the routing tables.

Pinging the Remote Addresses

Purpose Verify that the static routes are working.

From Device B, ping one of the loopback interface addresses on Device D.

From Device D, ping one of the loopback interface addresses on Device B.

Action

```
user@B> ping 192.168.47.5
PING 192.168.47.5 (192.168.47.5): 56 data bytes
64 bytes from 192.168.47.5: icmp_seq=0 ttl=64 time=156.126 ms
64 bytes from 192.168.47.5: icmp_seq=1 ttl=64 time=120.393 ms
64 bytes from 192.168.47.5: icmp_seq=2 ttl=64 time=175.361 ms

user@D> ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1): 56 data bytes
64 bytes from 10.0.0.1: icmp_seq=0 ttl=64 time=1.315 ms
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=31.819 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=1.268 ms
```

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Basic Static Routing on page 23
- Verifying the Static Route Configuration on page 89
- Example: Configuring IPv6 Static Routes on page 28

Example: Configuring IPv6 Static Routes

This example shows how to configure static routes when the interfaces have IPv6 addresses.

- Requirements on page 28
- Overview on page 28
- Configuration on page 29
- Verification on page 31

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

There are many practical applications for static routes. Static routing is often used at the network edge to support attachment to stub networks, which, given their single point of entry and egress, are well suited to the simplicity of a static route. In Junos OS, static routes have a global preference of 5. Static routes are activated if the specified next hop is reachable.

 Copyright © 2012, Juniper Networks, Inc.

In this example, you configure a static default route of ::/0, using a next-hop address 2001:db8:0:1:2a0:a502:0:1da.

For demonstration purposes, some loopback interfaces are configured on Device A and Device E. These loopback interfaces provide addresses to ping and thus verify that the static routes are working.

Figure 6 on page 29 shows the sample network.

Figure 6: Customer Routes Connected to a Service Provider

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device A

```
set interfaces fe-1/2/0 unit 1 description to-E
set interfaces fe-1/2/0 unit 1 family inet6 address 2001:db8:0:1:2a0:a502:0:1da/64
set interfaces lo0 unit 1 family inet6 address 2001:db8::1/128 primary
set interfaces lo0 unit 1 family inet6 address 2001:db8::2/128
set interfaces lo0 unit 1 family inet6 address 2001:db8::3/128
set routing-options rib inet6.0 static route 2001:db8::5/128 next-hop
  2001:db8:0:1:2a0:a502:0:19da
```

Device E

```
set interfaces fe-1/2/0 unit 25 description to-A
set interfaces fe-1/2/0 unit 25 family inet6 address 2001:db8:0:1:2a0:a502:0:19da/64
set interfaces lo0 unit 5 family inet6 address 2001:db8::5/128
set routing-options rib inet6.0 static route ::/0 next-hop 2001:db8:0:1:2a0:a502:0:1da
```

Copyright © 2012, Juniper Networks, Inc.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure basic static routes:

1. On Device A, configure the interfaces.

```
[edit interfaces]
set interfaces fe-1/2/0 unit 1 description to-E
set interfaces fe-1/2/0 unit 1 family inet6 address 2001:db8:0:1:2a0:a502:0:1da/64

set interfaces lo0 unit 1 family inet6 address 2001:db8::1/128 primary
set interfaces lo0 unit 1 family inet6 address 2001:db8::2/128
set interfaces lo0 unit 1 family inet6 address 2001:db8::3/128
```

2. On Device A, create a static route to Device E's loopback address and set the next-hop address.

   This ensures that Device A has a route back to Device E.

```
[edit routing-options]
set rib inet6.0 static route 2001:db8::5/128 next-hop 2001:db8:0:1:2a0:a502:0:19da
```

3. If you are done configuring Device A, commit the configuration.

```
[edit interfaces]
user@A# commit
```

4. On Device E, configure the interfaces.

```
[edit]
set interfaces fe-1/2/0 unit 25 description to-A
set interfaces fe-1/2/0 unit 25 family inet6 address 2001:db8:0:1:2a0:a502:0:19da/64
set interfaces lo0 unit 5 family inet6 address 2001:db8::5/128
```

5. On Device E, create a static default route and set the next-hop address.

```
[edit routing-options]
set routing-options rib inet6.0 static route ::/0 next-hop 2001:db8:0:1:2a0:a502:0:1da
```

6. If you are done configuring Device E, commit the configuration.

```
[edit]
user@E# commit
```

Results

Confirm your configuration by issuing the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device A

```
user@A# show interfaces
fe-1/2/0 {
   unit 1 {
      description to-E;
      family inet6 {
         address 2001:db8:0:1:2a0:a502:0:1da/64;
      }
   }
}
```


Copyright © 2012, Juniper Networks, Inc.

```
lo0 {
   unit 1 {
      family inet6 {
         address 2001:db8::1/128 {
            primary;
         }
         address 2001:db8::2/128;
         address 2001:db8::3/128;
      }
   }
}

user@A# show routing-options
rib inet6.0 {
   static {
      route 2001:db8::5/128 next-hop 2001:db8:0:1:2a0:a502:0:19da;
   }
}
```

Device E

```
user@E# show interfaces
fe-1/2/0 {
   unit 25 {
      description to-A;
      family inet6 {
         address 2001:db8:0:1:2a0:a502:0:19da/64;
      }
   }
}
lo0 {
   unit 5 {
      family inet6 {
         address 2001:db8::5/128;
      }
   }
}

user@E# show routing-options
rib inet6.0 {
   static {
      route ::/0 next-hop 2001:db8:0:1:2a0:a502:0:1da;
   }
}
```

Verification

Confirm that the configuration is working properly.

- Checking the Routing Tables on page 31
- Pinging the Remote Addresses on page 32

Checking the Routing Tables

Purpose Make sure that the static routes appear in the routing tables of Device A and Device E.

Action

```
user@A> show route protocol static
inet6.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8::5/128    *[Static/5] 00:27:46
                    > to 2001:db8:0:1:2a0:a502:0:19da via fe-1/2/0.1

user@E> show route protocol static
inet6.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

::/0               *[Static/5] 00:19:11
                    > to 2001:db8:0:1:2a0:a502:0:1da via fe-1/2/0.25
```

Meaning The static routes are in the routing tables.

Pinging the Remote Addresses

Purpose Verify that the static routes are working.

From Device A, ping one of the loopback interface addresses on Device E.

From Device E, ping one of the loopback interface addresses on Device A.

Action

```
user@A> ping 2001:db8::5
PING6(56=40+8+8 bytes) 2001:db8:0:1:2a0:a502:0:1da --> 2001:db8::5
16 bytes from 2001:db8::5, icmp_seq=0 hlim=64 time=1.790 ms
16 bytes from 2001:db8::5, icmp_seq=1 hlim=64 time=1.529 ms
16 bytes from 2001:db8::5, icmp_seq=2 hlim=64 time=1.531 ms

user@E> ping 2001:db8::3
PING6(56=40+8+8 bytes) 2001:db8:0:1:2a0:a502:0:19da --> 2001:db8::3
16 bytes from 2001:db8::3, icmp_seq=0 hlim=64 time=2.146 ms
16 bytes from 2001:db8::3, icmp_seq=1 hlim=64 time=1.964 ms
16 bytes from 2001:db8::3, icmp_seq=2 hlim=64 time=1.550 ms
```

Related Documentation

- Example: Configuring a Basic Set of Static Routes on page 24
- Example: Configuring Static Routes Between Logical Systems Within the Same Router on page 35

Static Routing on Logical Systems

- Example: Creating an Interface on a Logical System on page 33
- Example: Configuring Static Routes Between Logical Systems Within the Same Router on page 35
- Example: Connecting Logical Systems Within the Same Router Using Logical Tunnel Interfaces on page 40
- Example: Connecting a Logical System to a Physical Router on page 44

 Copyright © 2012, Juniper Networks, Inc.

Example: Creating an Interface on a Logical System

This example shows how to create an interface on a logical system.

- Requirements on page 33
- Overview on page 33
- Configuration on page 34
- Verification on page 35

Requirements

For the interface on the logical system to have connectivity, the corresponding physical interface must be administratively up, and the physical link must be up. You can verify the status of the physical interface by running the **show interfaces terse** command.

Overview

In logical systems, you must treat each interface like a point-to-point connection because you can only connect one logical tunnel interface to another at any given time. Also, you must select an interface encapsulation type, specify a DLCI number or VLAN identifier, configure a corresponding protocol family, and set the logical interface unit number of the peering **lt** interface.

To configure the interface encapsulation type, include the **dlci**, **encapsulation**, **family**, **peer-unit**, and **vlan-id** statements at the following hierarchy levels:

- M Series, MX Series, or T Series router (master administrator only)—**[edit interfaces lt-*fpc/pic/*0 unit *unit-number*]**
- Logical system—**[edit logical-systems *logical-system-name* interfaces *lt-fpc/pic/*0 unit *unit-number*]**

```
[edit]
logical-systems logical-system-name {
  interfaces {
    lt-fpc/pic/0 {
      unit unit-number {
        encapsulation (ethernet | ethernet-ccc | ethernet-vpls | frame-relay |
          frame-relay-ccc | vlan | vlan-ccc | vlan-vpls);
        peer-unit number; # The logical unit number of the peering lt interface.
        dlci dlci-number;
        vlan-id vlan-number;
        family (ccc | inet | inet6 | iso | mpls | tcc);
      }
    }
  }
}
```

NOTE: When you configure IPv6 addresses on a logical tunnel interface, you must configure unique IPv6 link-local addresses for any logical interfaces that peer with one another. To configure a link-local address, you must be the master administrator. Include a second IPv6 address with the **address** statement at the **[edit interfaces lt-*fpc/pic/port* unit *unit-number* family inet6]** hierarchy level. Link-local addresses typically begin with the numbers fe80 (such as fe80::1111:1/64).

In this example, you create the **fe-1/1/3** physical interface on the main router. You can also add values for properties that you need to configure on the physical interface, such as physical encapsulation, VLAN tagging (enabling), and link speed.

The example then shows how to assign logical interfaces to a logical system. Once you do this, the logical interfaces are considered part of the logical system.

Any logical interface unit can only be assigned to one system, including the main router. For example, if you configure logical unit 3 in the main router, you cannot configure logical unit 3 in a logical system.

In this example, you create logical unit 0 on Logical System LS1. You can also add values for properties that you need to configure on the logical interface, such as logical interface encapsulation, VLAN ID number, and protocol family.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set interfaces fe-1/1/3 description "main router interface"
set logical-systems LS1 interfaces fe-1/1/3 unit 0 description "LS1 interface"
set logical-systems LS1 interfaces fe-1/1/3 unit 0 family inet address 10.11.2.2/24
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure an interface on a logical system:

1. As the master administrator, configure the physical interface on the main router.

```
[edit]
user@host# set interfaces fe-1/1/3 description "main router interface"
```

2. Create the logical system interface on the logical unit.

```
[edit]
user@host# set logical-systems LS1 interfaces fe-1/1/3 unit 0 description "LS1
  interface"
user@host# set logical-systems LS1 interfaces fe-1/1/3 unit 0 family inet address
  10.11.2.2/24
```


Copyright © 2012, Juniper Networks, Inc.

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify that the configuration is working properly, issue the **show interfaces** command.

Related Documentation

- **ping** in the *Junos OS System Basics and Services Command Reference*
- **show interfaces detail** in the *Junos OS Interfaces Command Reference*

Example: Configuring Static Routes Between Logical Systems Within the Same Router

This example shows how to configure static routes between logical systems. The logical systems are configured in a single physical router and are connected by logical tunnel interfaces.

- Requirements on page 35
- Overview on page 35
- Configuration on page 36
- Verification on page 39

Requirements

You must connect the logical systems by using logical tunnel (**lt**) interfaces. See "Example: Connecting Logical Systems Within the Same Router Using Logical Tunnel Interfaces" on page 40.

Overview

A static route is a hard-coded path in the device that specifies how the route gets to a certain subnet by using a certain path. Routers that are connected to stub networks are often configured to use static routes. A *stub network* is a network with no knowledge of other networks. Stub networks send non-local traffic by way of a single path, with the network aware only of a default route to non-local destinations. In this example, you configure Logical System LS1 with a static route to the 10.10.10.0/30 network and define the next-hop address as 192.168.10.2. You also configure Logical System LS1 with a static route to the 192.168.10.0/30 network and define a next-hop address of 10.10.10.1.

Figure 7 on page 36 shows the sample network.

Figure 7: Static Routes Between Logical Systems

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set logical-systems LS1 interfaces lt-1/2/0 unit 0 description LS1->LS2
set logical-systems LS1 interfaces lt-1/2/0 unit 0 encapsulation ethernet
set logical-systems LS1 interfaces lt-1/2/0 unit 0 peer-unit 1
set logical-systems LS1 interfaces lt-1/2/0 unit 0 family inet address 192.168.10.1/30
set logical-systems LS2 interfaces lt-1/2/0 unit 1 description LS2->LS1
set logical-systems LS2 interfaces lt-1/2/0 unit 1 encapsulation ethernet
set logical-systems LS2 interfaces lt-1/2/0 unit 1 peer-unit 0
set logical-systems LS2 interfaces lt-1/2/0 unit 1 family inet address 192.168.10.2/30
set logical-systems LS2 interfaces lt-1/2/0 unit 9 description LS2->LS3
set logical-systems LS2 interfaces lt-1/2/0 unit 9 encapsulation ethernet
set logical-systems LS2 interfaces lt-1/2/0 unit 9 peer-unit 10
set logical-systems LS2 interfaces lt-1/2/0 unit 9 family inet address 10.10.10.1/30
set logical-systems LS3 interfaces lt-1/2/0 unit 10 description LS3->LS2
set logical-systems LS3 interfaces lt-1/2/0 unit 10 encapsulation ethernet
set logical-systems LS3 interfaces lt-1/2/0 unit 10 peer-unit 9
set logical-systems LS3 interfaces lt-1/2/0 unit 10 family inet address 10.10.10.2/30
set logical-systems LS1 routing-options static route 10.10.10.0/30 next-hop 192.168.10.2
set logical-systems LS3 routing-options static route 192.168.10.0/30 next-hop 10.10.10.1
```


Copyright © 2012, Juniper Networks, Inc.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure static routes between logical systems:

1. Run the **show interfaces terse** command to verify that the router has a logical tunnel (**lt**) interface.

```
user@host> show interfaces terse

Interface               Admin Link Proto    Local                 Remote
so-0/0/0                up    down
so-0/0/1                up    down
so-0/0/2                up    down
so-0/0/3                up    down
gr-1/2/0                up    up
ip-1/2/0                up    up
lt-1/2/0                up    up
...
```

2. Configure the logical tunnel interface on Logical System LS1 connecting to Logical System LS2.

```
[edit]
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 description LS1->LS2
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 encapsulation ethernet
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 peer-unit 1
user@host# set logical-systems LS1 interfaces lt-1/2/0 unit 0 family inet address
  192.168.10.1/30
```

3. Configure the logical tunnel interface on Logical System LS2 connecting to Logical System LS1.

```
[edit]
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 description LS2->LS1
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 encapsulation ethernet
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 peer-unit 0
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 1 family inet address
  192.168.10.2/30
```

4. Configure the logical tunnel interface on Logical System LS2 connecting to Logical System LS3.

```
[edit]
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 9 description LS2->LS3
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 9 encapsulation ethernet
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 9 peer-unit 10
user@host# set logical-systems LS2 interfaces lt-1/2/0 unit 9 family inet address
  10.10.10.1/30
```

5. Configure the logical tunnel interface on Logical System LS3 connecting to Logical System LS2.

```
[edit]
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 10 description LS3->LS2
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 10 encapsulation
  ethernet
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 10 peer-unit 9
```

Copyright © 2012, Juniper Networks, Inc.

```
user@host# set logical-systems LS3 interfaces lt-1/2/0 unit 10 family inet address
  10.10.10.2/30
```

6. Configure the static route on Logical System LS1 connecting to the 10.10.10.0/30 network.

```
[edit]
user@host# set logical-systems LS1 routing-options static route 10.10.10.0/30
  next-hop 192.168.10.2
```

7. Configure the static route on Logical System LS3 connecting to the 192.168.10.0/30 network.

```
[edit]
user@host# set logical-systems LS3 routing-options static route 192.168.10.0/30
  next-hop 10.10.10.1
```

8. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results Confirm your configuration by issuing the **show logical-systems** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show logical-systems
LS1 {
  interfaces {
    lt-1/2/0 {
      unit 0 {
        description LS1->LS2;
        encapsulation ethernet;
        peer-unit 1;
        family inet {
          address 192.168.10.1/30;
        }
      }
    }
  }
  routing-options {
    static {
      route 10.10.10.0/30 next-hop 192.168.10.2;
    }
  }
}
LS2 {
  interfaces {
    lt-1/2/0 {
      unit 1 {
        description LS2->LS1;
        encapsulation ethernet;
        peer-unit 0;
        family inet {
          address 192.168.10.2/30;
        }
      }
```

 Copyright © 2012, Juniper Networks, Inc.

```
            unit 9 {
              description LS2->LS3;
              encapsulation ethernet;
              peer-unit 10;
              family inet {
                address 10.10.10.1/30;
              }
            }
          }
        }
      }
      LS3 {
        interfaces {
          lt-1/2/0 {
            unit 10 {
              description LS3->LS2;
              encapsulation ethernet;
              peer-unit 9;
              family inet {
                address 10.10.10.2/30;
              }
            }
          }
        }
        routing-options {
          static {
            route 192.168.10.0/30 next-hop 10.10.10.1;
          }
        }
      }
```

Verification

Confirm that the configuration is working properly.

- Verifying That the Logical Systems Are Up on page 39
- Verifying Connectivity Between the Logical Systems on page 39

Verifying That the Logical Systems Are Up

Purpose Make sure that the interfaces are properly configured.

Action

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
...
lt-1/2/0                up    up
lt-1/2/0.0              up    up   inet     192.168.10.1/30
lt-1/2/0.1              up    up   inet     192.168.10.2/30
lt-1/2/0.9              up    up   inet     10.10.10.1/30
lt-1/2/0.10             up    up   inet     10.10.10.2/30
...
```

Verifying Connectivity Between the Logical Systems

Purpose Make sure that the static routes appear in the routing tables of Logical Systems LS1 and LS3. Also, make sure that the logical systems can ping each other.

Action

```
user@host> show route logical-system LS1
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30      *[Static/5] 18:43:25
                    > to 192.168.10.2 via lt-1/2/0.0
192.168.10.0/30    *[Direct/0] 18:43:25
                    > via lt-1/2/0.0
192.168.10.1/32    *[Local/0] 18:43:25
                       Local via lt-1/2/0.0

user@host> show route logical-system LS3
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30      *[Direct/0] 23:11:21
                    > via lt-1/2/0.10
10.10.10.2/32      *[Local/0] 23:11:21
                       Local via lt-1/2/0.10
192.168.10.0/30    *[Static/5] 00:23:31
                    > to 10.10.10.1 via lt-1/2/0.10
```

From LS1, Ping LS3

```
user@host> set cli logical-system LS1

user@host:LS1> ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2): 56 data bytes
64 bytes from 10.10.10.2: icmp_seq=0 ttl=63 time=1.263 ms
64 bytes from 10.10.10.2: icmp_seq=1 ttl=63 time=1.086 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=63 time=1.077 ms
```

From LS3, Ping LS1

```
user@host> set cli logical-system LS3

user@host:LS3> ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1): 56 data bytes
64 bytes from 192.168.10.1: icmp_seq=0 ttl=63 time=10.781 ms
64 bytes from 192.168.10.1: icmp_seq=1 ttl=63 time=1.167 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=63 time=1.152 ms
```

Related Documentation

- Example: Creating an Interface on a Logical System on page 33
- Example: Connecting Logical Systems Within the Same Router Using Logical Tunnel Interfaces on page 40

Example: Connecting Logical Systems Within the Same Router Using Logical Tunnel Interfaces

This example shows how to configure logical tunnel interfaces to connect two logical systems that are configured in a single router.

- Requirements on page 41
- Overview on page 41
- Configuration on page 42
- Verification on page 43

Copyright © 2012, Juniper Networks, Inc.

Requirements

On M Series and T Series routers, you can create a logical tunnel interface if you have a Tunnel Services PIC installed on an Enhanced FPC in your routing platform.

On M40e routers, you can create a logical tunnel interface if you have a Tunnel Services PIC. (An Enhanced FPC is not required.)

On an M7i router, logical tunnel interfaces can be created by using the integrated Adaptive Services Module.

On an MX Series router, the master administrator can configure logical tunnel interfaces by including the **tunnel-services** statement at the **[edit chassis fpc** *slot-number* **pic** *number***]** hierarchy level.

Overview

To connect two logical systems, you configure a logical tunnel interface on both logical systems. Then you configure a peer relationship between the logical tunnel interfaces, thus creating a point-to-point connection. Logical tunnel interfaces behave like regular interfaces. You can configure them with Ethernet, Frame Relay, or another encapsulation type. You can also configure routing protocols across them. In effect, the logical tunnel (**lt**) interfaces connect two logical systems within the same router. The two logical systems do not share routing tables. This means that you can run dynamic routing protocols between different logical systems within the same router.

You must treat each interface like a point-to-point connection because you can only connect one logical tunnel interface to another at any given time. Also, you must select an interface encapsulation type, configure a corresponding protocol family, and set the logical interface unit number of the peering **lt** interface.

In this example, the logical tunnel interfaces are configured to behave as Ethernet interfaces with the **encapsulation ethernet** statement. The IS-IS Protocol is enabled on the logical tunnel interfaces with the **family iso** statement.

When configuring logical tunnel interfaces, note the following:

- The peering logical interfaces must have the same physical **lt** interface name. For example, a logical unit on **lt-0/1/0** cannot peer with a logical unit on **lt-0/0/10**. The FPC, PIC, and port numbers must match.
- The peering logical interfaces must be derived from the same PIC or module.
- You can configure only one peer unit for each logical interface. For example, unit 0 cannot peer with both unit 1 and unit 2.
- Logical tunnels are not supported with Adaptive Services, MultiServices, or Link Services PICs, but they are supported on the Adaptive Services Module on M7i routers.

Figure 8 on page 42 shows the topology used in this example.

Copyright © 2012, Juniper Networks, Inc.

Figure 8: Connecting Two Logical Systems

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set logical-systems LS1 interfaces lt-0/1/0 unit 0 description LS1->LS2
set logical-systems LS1 interfaces lt-0/1/0 unit 0 encapsulation ethernet
set logical-systems LS1 interfaces lt-0/1/0 unit 0 peer-unit 1
set logical-systems LS1 interfaces lt-0/1/0 unit 0 family inet address 10.0.8.13/30
set logical-systems LS1 interfaces lt-0/1/0 unit 0 family iso
set logical-systems LS2 interfaces lt-0/1/0 unit 1 description LS2->LS1
set logical-systems LS2 interfaces lt-0/1/0 unit 1 encapsulation ethernet
set logical-systems LS2 interfaces lt-0/1/0 unit 1 peer-unit 0
set logical-systems LS2 interfaces lt-0/1/0 unit 1 family inet address 10.0.8.14/30
set logical-systems LS2 interfaces lt-0/1/0 unit 1 family iso
```


Copyright © 2012, Juniper Networks, Inc.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To connect logical system interfaces:

1. Run the **show interfaces terse** command to verify that the router has a logical tunnel (**lt**) interface.

```
user@host> show interfaces terse

Interface               Admin Link Proto    Local                 Remote
so-0/0/0                up    down
so-0/0/1                up    down
so-0/0/2                up    down
so-0/0/3                up    down
gr-0/1/0                up    up
ip-0/1/0                up    up
lt-0/1/0                up    up
...
```

2. Configure the logical tunnel interface on Logical System LS1.

```
[edit]
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 description LS1->LS2
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 encapsulation ethernet
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 peer-unit 1
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 family inet address
  10.0.8.13/30
user@host# set logical-systems LS1 interfaces lt-0/1/0 unit 0 family iso
```

3. Configure the logical tunnel interface on Logical System LS2.

```
[edit]
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 description LS2->LS1
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 encapsulation ethernet
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 peer-unit 0
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 family inet address
  10.0.8.14/30
user@host# set logical-systems LS2 interfaces lt-0/1/0 unit 1 family iso
```

4. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

Confirm that the configuration is working properly.

- Verifying That the Logical Systems Are Up on page 43
- Verifying Connectivity Between the Logical Systems on page 44

Verifying That the Logical Systems Are Up

Purpose Make sure that the interfaces are properly configured.

Action

```
user@host> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
so-0/0/0                up    down
so-0/0/1                up    down
so-0/0/2                up    down
so-0/0/3                up    down
gr-0/1/0                up    up
ip-0/1/0                up    up
lt-0/1/0                up    up
lt-0/1/0.0              up    up   inet     10.0.8.13/30
                                   iso
lt-0/1/0.1              up    up   inet     10.0.8.14/30
                                   iso
...
```

Verifying Connectivity Between the Logical Systems

Purpose Make sure that the network address appears as directly connected.

Action

```
user@host> show route logical-system all
logical-system: LS1

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.8.12/30       *[Direct/0] 00:00:34
                    > via lt-0/1/0.0
10.0.8.13/32       *[Local/0] 00:00:34
                      Local via lt-0/1/0.0
-----

logical-system: LS2

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.8.12/30       *[Direct/0] 00:00:34
                    > via lt-0/1/0.1
10.0.8.14/32       *[Local/0] 00:00:34
                      Local via lt-0/1/0.1
...
```

Related Documentation

- Example: Connecting a Logical System to a Physical Router on page 44
- Example: Creating an Interface on a Logical System on page 33

Example: Connecting a Logical System to a Physical Router

This example shows how to configure an interface on a logical system to connect to a separate router. The separate router can be a physical router or a logical system on a physical router.

- Requirements on page 45
- Overview on page 45
- Configuration on page 45
- Verification on page 46

 Copyright © 2012, Juniper Networks, Inc.

Requirements

PICs must be installed on the two routers.

Overview

In this example, Logical System LS1 is configured on Router R1. The Logical System LS1 has a direct connection to Router R2.

Figure 9 on page 45 shows the topology used in this example.

Figure 9: Logical System Connected to a Physical Router

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Router R1

```
set interfaces so-0/0/2 description "main router interface to R2"
set logical-systems LS1 interfaces so-0/0/2 unit 0 description LS1->R2
set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet address 10.0.45.2/30
```

Device R2

```
set interfaces so-0/0/2 description R2->LS1
set interfaces so-0/0/2 unit 0 family inet address 10.0.45.1/30
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To connect a logical system to a physical router:

1. On Router R1, configure the interface.

```
[edit]
user@R1# set interfaces so-0/0/2 description "main router interface to R2"
```

2. On Router R1, configure the Logical System LS1 interface.

```
[edit]
user@R1# set logical-systems LS1 interfaces so-0/0/2 unit 0 description LS1->R2
user@R1# set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet address
  10.0.45.2/30
```

3. On Device R2, configure the interface to Logical System LS1.

```
[edit]
user@R2# set interfaces so-0/0/2 description R2->LS1
user@R2# set interfaces so-0/0/2 unit 0 family inet address 10.0.45.1/30
```

4. If you are done configuring the devices, commit the configurations.

```
[edit]
user@host# commit
```

Verification

Confirm that the configuration is working properly.

Verifying Connectivity

Purpose Make sure that the devices can ping each other.

Action

```
user@R2> ping 10.0.45.2
PING 10.0.45.2 (10.0.45.2): 56 data bytes
64 bytes from 10.0.45.2: icmp_seq=0 ttl=64 time=3.910 ms
64 bytes from 10.0.45.2: icmp_seq=1 ttl=64 time=3.559 ms
64 bytes from 10.0.45.2: icmp_seq=2 ttl=64 time=3.503 ms

user@R1> set cli logical-system LS1
Logical system: LS1

user@R1:LS1> ping 10.0.45.1
PING 10.0.45.1 (10.0.45.1): 56 data bytes
64 bytes from 10.0.45.1: icmp_seq=0 ttl=64 time=1.217 ms
64 bytes from 10.0.45.1: icmp_seq=1 ttl=64 time=1.183 ms
64 bytes from 10.0.45.1: icmp_seq=2 ttl=64 time=1.121 ms
```

Related Documentation

- Example: Creating an Interface on a Logical System on page 33
- Example: Connecting Logical Systems Within the Same Router Using Logical Tunnel Interfaces on page 40

Static Route Selection

- Understanding Static Route Preferences and Qualified Next Hops on page 46
- Example: Configuring Static Route Preferences and Qualified Next Hops on page 47

Understanding Static Route Preferences and Qualified Next Hops

A static route destination address can have multiple next hops associated with it. In this case, multiple routes are inserted into the routing table, and route selection must occur.

 Copyright © 2012, Juniper Networks, Inc.

Because the primary criterion for route selection is the route preference, you can control the routes that are used as the primary route for a particular destination by setting the route preference associated with a particular next hop. The routes with a lower route preference are always used to route traffic. When you do not set a preferred route, traffic is alternated between routes in round-robin fashion.

In general, the default properties assigned to a static route apply to all the next-hop addresses configured for the static route. If, however, you want to configure two possible next-hop addresses for a particular route and have them treated differently, you can define one as a qualified next hop.

Qualified next hops allow you to associate one or more properties with a particular next-hop address. You can set an overall preference for a particular static route and then specify a different preference for the qualified next hop. For example, suppose two next-hop addresses (10.10.10.10 and 10.10.10.7) are associated with the static route 192.168.47.5/32. A general preference is assigned to the entire static route, and then a different preference is assigned to only the qualified next-hop address 10.10.10.7. For example:

```
route 192.168.47.5/32 {
   next-hop 10.10.10.10;
   qualified-next-hop 10.10.10.7 {
      preference 2;
   }
   preference 6;
}
```

In this example, the qualified next hop 10.10.10.7 is assigned the preference 2, and the next-hop 10.10.10.10 is assigned the preference 6.

NOTE: The preference and metric options only apply to the qualified next hops. The qualified next-hop preference and metric override the route preference and metric (for that specific qualified next hop), similar to how the route preference overrides the default preference and metric (for that specific route).

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring Static Route Preferences and Qualified Next Hops on page 47

Example: Configuring Static Route Preferences and Qualified Next Hops

This example shows how to control static route selection.

- Requirements on page 48
- Overview on page 48
- Configuration on page 48
- Verification on page 51

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

In this example, the static route 192.168.47.0/24 has two possible next hops. Because one link has higher bandwidth, this link is the preferred path. To enforce this preference, the **qualified-next-hop** statement is included in the configuration on both devices. See Figure 10 on page 48.

Figure 10: Controlling Static Route Selection

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device B

```
set interfaces ge-1/2/0 unit 0 description B->D
set interfaces ge-1/2/0 unit 0 family inet address 172.16.1.1/24
set interfaces fe-1/2/1 unit 2 description secondary-B->D
set interfaces fe-1/2/1 unit 2 family inet address 192.168.2.1/24
set interfaces lo0 unit 57 family inet address 10.0.0.1/32
set interfaces lo0 unit 57 family inet address 10.0.0.2/32
set routing-options static route 192.168.47.0/24 next-hop 172.16.1.2
set routing-options static route 192.168.47.0/24 qualified-next-hop 192.168.2.2 preference
    25
```

Device D

```
set interfaces ge-1/2/0 unit 1 description D->B
set interfaces ge-1/2/0 unit 1 family inet address 172.16.1.2/24
set interfaces fe-1/2/1 unit 3 description secondary-D->B
set interfaces fe-1/2/1 unit 3 family inet address 192.168.2.2/24
```


Copyright © 2012, Juniper Networks, Inc.

```
set interfaces lo0 unit 2 family inet address 192.168.47.5/32
set interfaces lo0 unit 2 family inet address 192.168.47.6/32
set routing-options static route 0.0.0.0/0 next-hop 172.16.1.1
set routing-options static route 0.0.0.0/0 qualified-next-hop 192.168.2.1 preference 25
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To control static route selection:

1. On Device B, configure the interfaces.

```
[edit interfaces]
user@B# set ge-1/2/0 unit 0 description B->D
user@B# set ge-1/2/0 unit 0 family inet address 172.16.1.1/24
user@B# set fe-1/2/1 unit 2 description secondary-B->D
user@B# set fe-1/2/1 unit 2 family inet address 192.168.2.1/24
user@B# set lo0 unit 57 family inet address 10.0.0.1/32
user@B# set lo0 unit 57 family inet address 10.0.0.2/32
```

2. On Device B, configure a static route to the customer network.

```
[edit routing-options static route 192.168.47.0/24]
user@B# set next-hop 172.16.1.2
```

3. On Device B, configure a backup route to the customer network.

```
[edit routing options static route 192.168.47.0/24]
user@B# set qualified-next-hop 192.168.2.2 preference 25
```

4. On Device D, configure the interfaces.

```
[edit interfaces]
user@D# set ge-1/2/0 unit 1 description D->B
user@D# set ge-1/2/0 unit 1 family inet address 172.16.1.2/24
user@D# set fe-1/2/1 unit 3 description secondary-D->B
user@D# set fe-1/2/1 unit 3 family inet address 192.168.2.2/24
user@D# set lo0 unit 2 family inet address 192.168.47.5/32
user@D# set lo0 unit 2 family inet address 192.168.47.6/32
```

5. On Device D, configure a static default route to external networks.

```
[edit routing options static route 0.0.0.0/0]
user@D# set next-hop 172.16.1.1
```

6. On Device D, configure a backup static default route to external networks.

```
[edit routing options static route 0.0.0.0/0]
user@D# set qualified-next-hop 192.168.2.1 preference 25
```

Results

Confirm your configuration by issuing the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@B# show interfaces
ge-1/2/0 {
   unit 0 {
      description B->D;
```

```
        family inet {
            address 172.16.1.1/24;
        }
    }
}
fe-1/2/1 {
    unit 2 {
        description secondary-B->D;
        family inet {
            address 192.168.2.1/24;
        }
    }
}
lo0 {
    unit 57 {
        family inet {
            address 10.0.0.1/32;
            address 10.0.0.2/32;
        }
    }
}
```

```
user@B# show routing-options
static {
    route 192.168.47.0/24 {
        next-hop 172.16.1.2;
        qualified-next-hop 192.168.2.2 {
            preference 25;
        }
    }
}
```

```
user@D# show interfaces
ge-1/2/0 {
    unit 1 {
        description D->B;
        family inet {
            address 172.16.1.2/24;
        }
    }
}
fe-1/2/1 {
    unit 3 {
        description secondary-D->B;
        family inet {
            address 192.168.2.2/24;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 192.168.47.5/32;
            address 192.168.47.6/32;
        }
    }
}
```


Copyright © 2012, Juniper Networks, Inc.

```
user@D# show routing-options
static {
  route 0.0.0.0/0 {
    next-hop 172.16.1.1;
    qualified-next-hop 192.168.2.1 {
      preference 25;
    }
  }
}
```

If you are done configuring the devices, enter **commit** from configuration mode on both devices.

Verification

Confirm that the configuration is working properly.

- Checking the Routing Tables on page 51
- Pinging the Remote Addresses on page 51
- Making Sure That the Backup Route Becomes the Active Route on page 52

Checking the Routing Tables

Purpose Make sure that the static routes appear in the routing tables of Device B and Device D.

Action

```
user@B> show route protocol static
inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.47.0/24    *[Static/5] 02:02:03
                    > to 172.16.1.2 via ge-1/2/0.0
                    [Static/25] 01:58:21
                    > to 192.168.2.2 via fe-1/2/1.2

user@D> show route protocol static
inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 02:02:12
                    > to 172.16.1.1 via ge-1/2/0.1
                    [Static/25] 01:58:31
                    > to 192.168.2.1 via fe-1/2/1.3
```

Meaning The asterisks (*) in the routing tables show the active routes. The backup routes are listed next.

Pinging the Remote Addresses

Purpose Verify that the static routes are working.

From Device B, ping one of the loopback interface addresses on Device D.

From Device D, ping one of the loopback interface addresses on Device B.

Action

```
user@B> ping 192.168.47.5
PING 192.168.47.5 (192.168.47.5): 56 data bytes
64 bytes from 192.168.47.5: icmp_seq=0 ttl=64 time=156.126 ms
64 bytes from 192.168.47.5: icmp_seq=1 ttl=64 time=120.393 ms
64 bytes from 192.168.47.5: icmp_seq=2 ttl=64 time=175.361 ms

user@D> ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1): 56 data bytes
64 bytes from 10.0.0.1: icmp_seq=0 ttl=64 time=1.315 ms
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=31.819 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=1.268 ms
```

Making Sure That the Backup Route Becomes the Active Route

Purpose If the primary route becomes unusable, make sure that the backup secondary route becomes active.

Action

1. Disable the active route by deactivating the ge-1/2/0.0 interface on Device B.

   user@B# deactivate interfaces ge-1/2/0 unit 0 family inet address 172.16.1.1/24
   user@B# commit

2. Check Device B's routing table.

```
user@B> show route protocol static

inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.47.0/24    *[Static/25] 02:06:24
                    > to 192.168.2.2 via fe-1/2/1.2
```

Meaning The backup route has become the active route.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Static Route Preferences and Qualified Next Hops on page 46
- Verifying the Static Route Configuration on page 89

Static Route Control in Routing and Forwarding Tables

- Understanding Static Route Control in Routing and Forwarding Tables on page 52
- Example: Preventing a Static Route from Being Readvertised on page 53

Understanding Static Route Control in Routing and Forwarding Tables

You can control the importation of static routes into the routing and forwarding tables in a number of ways. Primary ways include assigning one or more of the following attributes to the route:

- **retain**—Keeps the route in the forwarding table after the routing process shuts down or the device reboots.
- **no-readvertise**—Prevents the route from being readvertised to other routing protocols.

 Copyright © 2012, Juniper Networks, Inc.

- **passive**—Rejects traffic destined for the route.

This topic includes the following sections:

- Route Retention on page 53
- Readvertisement Prevention on page 53
- Forced Rejection of Passive Route Traffic on page 53

Route Retention

By default, static routes are not retained in the forwarding table when the routing process shuts down. When the routing process starts up again, any routes configured as static routes must be added to the forwarding table again. To avoid this latency, routes can be flagged as **retain**, so that they are kept in the forwarding table even after the routing process shuts down. Retention ensures that the routes are always in the forwarding table, even immediately after a system reboot.

Readvertisement Prevention

Static routes are eligible for readvertisement by other routing protocols by default. In a stub area where you might not want to readvertise these static routes under any circumstances, you can flag the static routes as **no-readvertise**.

Forced Rejection of Passive Route Traffic

Generally, only active routes are included in the routing and forwarding tables. If a static route's next-hop address is unreachable, the route is marked **passive**, and it is not included in the routing or forwarding tables. To force a route to be included in the routing tables regardless of next-hop reachability, you can flag the route as **passive**. If a route is flagged **passive** and its next-hop address is unreachable, the route is included in the routing table, and all traffic destined for the route is rejected.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Preventing a Static Route from Being Readvertised on page 53

Example: Preventing a Static Route from Being Readvertised

This example shows how to prevent a static route from being readvertised into OSPF, thereby preventing the route from appearing in the routing and forwarding tables.

- Requirements on page 53
- Overview on page 54
- Configuration on page 54
- Verification on page 58

Requirements

In this example, no special configuration beyond device initialization is required.

Copyright © 2012, Juniper Networks, Inc.

Overview

This example shows how to configure a routing policy that readvertises static routes into OSPF, with the exception of one static route that is not readvertised because it is tagged with the **no-readvertise** statement.

Figure 11 on page 54 shows the sample network.

Figure 11: Customer Routes Connected to a Service Provider

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device A

```
set interfaces fe-1/2/0 unit 4 description A->B
set interfaces fe-1/2/0 unit 4 family inet address 10.0.2.2/30
set protocols ospf area 0.0.0.0 interface fe-1/2/0.4
```

Device B

```
set interfaces fe-1/2/0 unit 3 description B->A
set interfaces fe-1/2/0 unit 3 family inet address 10.0.2.1/30
set interfaces fe-1/2/1 unit 6 description B->C
set interfaces fe-1/2/1 unit 6 family inet address 10.0.3.1/30
set protocols bgp group ext type external
set protocols bgp group ext peer-as 23
set protocols bgp group ext neighbor 10.0.3.2
set protocols ospf export send-static
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set policy-options policy-statement send-static from protocol static
set policy-options policy-statement send-static then accept
set routing-options static route 0.0.0.0/0 next-hop 10.0.3.2
set routing-options static route 192.168.0.0/24 next-hop 10.0.3.2
set routing-options static route 192.168.0.0/24 no-readvertise
set routing-options autonomous-system 17
```


Copyright © 2012, Juniper Networks, Inc.

Device C

```
set interfaces fe-1/2/0 unit 7 description B->C
set interfaces fe-1/2/0 unit 7 family inet address 10.0.3.2/30
set interfaces lo0 unit 5 family inet address 192.168.0.1/32
set protocols bgp group ext type external
set protocols bgp group ext peer-as 17
set protocols bgp group ext neighbor 10.0.3.1
set routing-options autonomous-system 23
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device A:

1. Configure the interface to Device B.

```
[edit interfaces fe-1/2/0 unit 4]
user@A# set description A->B
user@A# set family inet address 10.0.2.2/30
```

2. Configure OSPF to form an OSPF peer relationship with Device B.

```
[edit protocols ospf area 0.0.0.0]
user@A# set interface fe-1/2/0.4
```

Step-by-Step Procedure

To configure Device B:

1. Configure the interfaces to Device A and Device C.

```
[edit interfaces]
user@B# set fe-1/2/0 unit 3 description B->A
user@B# set fe-1/2/0 unit 3 family inet address 10.0.2.1/30
user@B# set fe-1/2/1 unit 6 description B->C
user@B# set fe-1/2/1 unit 6 family inet address 10.0.3.1/30
```

2. Configure one or more static routes and the autonomous system (AS) number.

```
[edit routing-options]
user@B# set static route 0.0.0.0/0 next-hop 10.0.3.2
user@B# set static route 192.168.0.0/24 next-hop 10.0.3.2
user@B# set autonomous-system 17
```

3. Configure the routing policy.

   This policy exports static routes from the routing table into OSPF.

```
[edit policy-options policy-statement send-static]
user@B# set from protocol static
user@B# set then accept
```

4. Include the **no-readvertise** statement to prevent the 192.168.0.0/24 route from being exported into OSPF.

```
[edit routing-options]
user@B# set static route 192.168.0.0/24 no-readvertise
```

5. Configure the routing protocols.

The BGP configuration forms an external BGP (EBGP) peer relationship with Device C.

The OSPF configuration forms an OSPF peer relationship with Device A and applies the **send-static** routing policy.

```
[edit protocols]
user@B# set bgp group ext type external
user@B# set bgp group ext peer-as 23
user@B# set bgp group ext neighbor 10.0.3.2
user@B# set ospf export send-static
user@B# set ospf area 0.0.0.0 interface fe-1/2/0.3
```

Step-by-Step Procedure

To configure Device C:

1. Create the interface to Device B, and configure the loopback interface.

```
[edit interfaces ]
user@C# set fe-1/2/0 unit 7 description B->C
user@C# set fe-1/2/0 unit 7 family inet address 10.0.3.2/30
user@C# set lo0 unit 5 family inet address 192.168.0.1/32
```

2. Configure the EBGP peering session with Device B.

```
[edit protocols bgp group ext]
user@C# set type external
user@C# set peer-as 17
user@C# set neighbor 10.0.3.1
```

3. Configure the AS number.

```
[edit routing-options]
user@C# set autonomous-system 23
```

Results

Confirm your configuration by issuing the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device A

```
user@A# show interfaces
fe-1/2/0 {
   unit 4 {
      description A->B;
      family inet {
         address 10.0.2.2/30;
      }
   }
}

user@A# show protocols
ospf {
   area 0.0.0.0 {
      interface fe-1/2/0.4;
   }
}
```

Device B

```
user@B# show interfaces
interfaces {
```


Copyright © 2012, Juniper Networks, Inc.

```
    fe-1/2/0 {
      unit 3 {
        description B->A;
        family inet {
          address 10.0.2.1/30;
        }
      }
    }
    fe-1/2/1 {
      unit 6 {
        description B->C;
        family inet {
          address 10.0.3.1/30;
        }
      }
    }
  }

  user@B# show policy-options
  policy-statement send-static {
    from protocol static;
    then accept;
  }

  user@B# show protocols
  bgp {
    group ext {
      type external;
      peer-as 23;
      neighbor 10.0.3.2;
    }
  }
  ospf {
    export send-static;
    area 0.0.0.0 {
      interface fe-1/2/0.3;
    }
  }

  user@B# show routing-options
  static {
    route 0.0.0.0/0 next-hop 10.0.3.2;
    route 192.168.0.0/24 {
      next-hop 10.0.3.2;
      no-readvertise;
    }
  }
  autonomous-system 17;
```

Device C

```
  user@C# show interfaces
  fe-1/2/0 {
    unit 7 {
      description B->C;
      family inet {
        address 10.0.3.2/30;
      }
    }
```

```
}
lo0 {
  unit 5 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}

user@C# show protocols
bgp {
  group ext {
    type external;
    peer-as 17;
    neighbor 10.0.3.1;
  }
}

user@C# show routing-options
autonomous-system 23;
```

If you are done configuring the devices, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Checking the Routing Table

Purpose Make sure that the **no-readvertise** statement is working.

Action

1. On Device A, run the **show route protocol ospf** command to make sure that the 192.168.0.0/24 route does not appear in Device A's routing table.

```
user@A> show route protocols ospf

inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[OSPF/150] 00:03:15, metric 0, tag 0
                    > to 10.0.2.1 via fe-1/2/0.4
224.0.0.5/32       *[OSPF/10] 00:04:07, metric 1
                      MultiRecv
```

2. On Device B, deactivate the **no-readvertise** statement.

```
user@B# deactivate routing-options static route 192.168.0.0/24 no-readvertise
```

3. On Device A, rerun the **show route protocol ospf** command to make sure that the 192.168.0.0/24 route appears in Device A's routing table.

```
user@A> show route protocols ospf

inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[OSPF/150] 00:04:24, metric 0, tag 0
                    > to 10.0.2.1 via fe-1/2/0.4
192.168.0.0/24   *[OSPF/150] 00:00:15, metric 0, tag 0
```

 Copyright © 2012, Juniper Networks, Inc.

```
                > to 10.0.2.1 via fe-1/2/0.4
224.0.0.5/32       *[OSPF/10] 00:05:16, metric 1
                      MultiRecv
```

Meaning The **no-readvertise** statement is working as expected.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Static Route Control in Routing and Forwarding Tables on page 52
- Verifying the Static Route Configuration on page 89

Static Routes and Point-to-Multipoint LSPs

- Understanding Point-to-Multipoint LSPs on page 59
- Point-to-Multipoint LSP Configuration Overview on page 60
- Example: Configuring an RSVP-Signaled Point-to-Multipoint LSP on page 61

Understanding Point-to-Multipoint LSPs

A point-to-multipoint MPLS label-switched path (LSP) is a Resource Reservation Protocol (RSVP) signaled-signaled LSP with a single source and multiple destinations. By taking advantage of the MPLS packet replication capability of the network, point-to-multipoint LSPs avoid unnecessary packet replication at the inbound (ingress) router. Packet replication takes place only when packets are forwarded to two or more different destinations requiring different network paths.

This process is illustrated in Figure 12 on page 60. Router PE1 is configured with a point-to-multipoint LSP to Routers PE2, PE3, and PE4. When Router PE1 sends a packet on the point-to-multipoint LSP to Routers P1 and P2, Router P1 replicates the packet and forwards it to Routers PE2 and PE3. Router P2 sends the packet to Router PE4.

Figure 12: Point-to-Multipoint LSPs

The following are some of the properties of point-to-multipoint LSPs:

- A point-to-multipoint LSP allows you to use MPLS for point-to-multipoint data distribution. This functionality is similar to that provided by IP multicast.
- You can add and remove branch LSPs from a main point-to-multipoint LSP without disrupting traffic. The unaffected parts of the point-to-multipoint LSP continue to function normally.
- You can configure a node to be both a transit and an outbound (egress) router for different branch LSPs of the same point-to-multipoint LSP.
- You can enable link protection on a point-to-multipoint LSP. Link protection can provide a bypass LSP for each of the branch LSPs that make up the point-to-multipoint LSP. If any primary paths fail, traffic can be quickly switched to the bypass.
- You can configure sub-paths either statically or dynamically.
- You can enable graceful restart on point-to-multipoint LSPs.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Point-to-Multipoint LSP Configuration Overview on page 60

Point-to-Multipoint LSP Configuration Overview

To set up a point-to-multipoint LSP:

1. Configure the primary LSP from the ingress router and the branch LSPs that carry traffic to the egress routers.

Copyright © 2012, Juniper Networks, Inc.

2. Specify a pathname on the primary LSP and this same path name on each branch LSP.

NOTE: By default, the branch LSPs are dynamically signaled by means of Constrained Shortest Path First (CSPF) and require no configuration. You can alternatively configure the branch LSPs as static paths.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Point-to-Multipoint LSPs on page 59
- *Junos OS MPLS Applications Configuration Guide*

Example: Configuring an RSVP-Signaled Point-to-Multipoint LSP

This example shows how to configure a collection of paths to create an RSVP-signaled point-to-multipoint label-switched path (LSP).

- Requirements on page 61
- Overview on page 61
- Configuration on page 62
- Verification on page 78

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

In this example, multiple routing devices serve as the transit, branch, and leaf nodes of a single point-to-multipoint LSP. On the provider edge (PE), Device PE1 is the ingress node. The branches go from PE1 to PE2, PE1 to PE3, and PE1 to PE4. Static unicast routes on the ingress node (PE1) point to the egress nodes.

This example also demonstrates static routes with a next hop that is a point-to-multipoint LSP, using the **p2mp-lsp-next-hop** statement. This is useful when implementing filter-based forwarding.

NOTE: Another option is to use the **lsp-next-hop** statement to configure a regular point-to-point LSP to be the next hop. Though not shown in this example, you can optionally assign an independent preference and metric to the next hop.

Topology Diagram

Figure 13 on page 62 shows the topology used in this example.

Figure 13: RSVP-Signaled Point-to-Multipoint LSP

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device PE1

```
set interfaces ge-2/0/2 unit 0 description PE1-to-CE1
set interfaces ge-2/0/2 unit 0 family inet address 10.0.244.10/30
set interfaces fe-2/0/10 unit 1 description PE1-to-P2
set interfaces fe-2/0/10 unit 1 family inet address 2.2.2.1/24
set interfaces fe-2/0/10 unit 1 family mpls
set interfaces fe-2/0/9 unit 8 description PE1-to-P3
set interfaces fe-2/0/9 unit 8 family inet address 6.6.6.1/24
set interfaces fe-2/0/9 unit 8 family mpls
set interfaces fe-2/0/8 unit 9 description PE1-to-P4
set interfaces fe-2/0/8 unit 9 family inet address 3.3.3.1/24
set interfaces fe-2/0/8 unit 9 family mpls
set interfaces lo0 unit 1 family inet address 100.10.10.10/32
set protocols rsvp interface fe-2/0/10.1
set protocols rsvp interface fe-2/0/9.8
set protocols rsvp interface fe-2/0/8.9
set protocols rsvp interface lo0.1
set protocols mpls traffic-engineering bgp-igp
set protocols mpls label-switched-path PE1-PE2 to 100.50.50.50
set protocols mpls label-switched-path PE1-PE2 link-protection
set protocols mpls label-switched-path PE1-PE2 p2mp p2mp1
set protocols mpls label-switched-path PE1-PE3 to 100.70.70.70
set protocols mpls label-switched-path PE1-PE3 link-protection
set protocols mpls label-switched-path PE1-PE3 p2mp p2mp1
set protocols mpls label-switched-path PE1-PE4 to 100.40.40.40
set protocols mpls label-switched-path PE1-PE4 link-protection
```


Copyright © 2012, Juniper Networks, Inc.

```
set protocols mpls label-switched-path PE1-PE4 p2mp p2mp1
set protocols mpls interface fe-2/0/10.1
set protocols mpls interface fe-2/0/9.8
set protocols mpls interface fe-2/0/8.9
set protocols mpls interface lo0.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-2/0/2.0
set protocols ospf area 0.0.0.0 interface fe-2/0/10.1
set protocols ospf area 0.0.0.0 interface fe-2/0/9.8
set protocols ospf area 0.0.0.0 interface fe-2/0/8.9
set protocols ospf area 0.0.0.0 interface lo0.1
set routing-options static route 5.5.5.0/24 p2mp-lsp-next-hop p2mp1
set routing-options static route 7.7.7.0/24 p2mp-lsp-next-hop p2mp1
set routing-options static route 4.4.4.0/24 p2mp-lsp-next-hop p2mp1
set routing-options router-id 100.10.10.10
```

Device CE1

```
set interfaces ge-1/3/2 unit 0 family inet address 10.0.244.9/30
set interfaces ge-1/3/2 unit 0 description CE1-to-PE1
set routing-options static route 10.0.104.8/30 next-hop 10.0.244.10
set routing-options static route 10.0.134.8/30 next-hop 10.0.244.10
set routing-options static route 10.0.224.8/30 next-hop 10.0.244.10
```

Device CE2

```
set interfaces ge-1/3/3 unit 0 family inet address 10.0.224.9/30
set interfaces ge-1/3/3 unit 0 description CE2-to-PE2
set routing-options static route 10.0.244.8/30 next-hop 10.0.224.10
```

Device CE3

```
set interfaces ge-2/0/1 unit 0 family inet address 10.0.134.9/30
set interfaces ge-2/0/1 unit 0 description CE3-to-PE3
set routing-options static route 10.0.244.8/30 next-hop 10.0.134.10
```

Device CE4

```
set interfaces ge-3/1/3 unit 0 family inet address 10.0.104.10/30
set interfaces ge-3/1/3 unit 0 description CE4-to-PE4
set routing-options static route 10.0.244.8/30 next-hop 10.0.104.9
```

Configuring the Ingress Label-Switched Router (LSR) (Device PE1)

Step-by-Step Procedure

To configure Device PE1:

1. Configure the interfaces, interface encapsulation, and protocol families.

```
[edit interfaces]
user@PE1# set ge-2/0/2 unit 0 description PE1-to-CE1
user@PE1# set ge-2/0/2 unit 0 family inet address 10.0.244.10/30
user@PE1# set fe-2/0/10 unit 1 description PE1-to-P2
user@PE1# set fe-2/0/10 unit 1 family inet address 2.2.2.1/24
user@PE1# set fe-2/0/10 unit 1 family mpls
user@PE1# set fe-2/0/9 unit 8 description PE1-to-P3
user@PE1# set fe-2/0/9 unit 8 family inet address 6.6.6.1/24
user@PE1# set fe-2/0/9 unit 8 family mpls
user@PE1# set fe-2/0/8 unit 9 description PE1-to-P4
user@PE1# set fe-2/0/8 unit 9 family inet address 3.3.3.1/24
user@PE1# set fe-2/0/8 unit 9 family mpls
user@PE1# set lo0 unit 1 family inet address 100.10.10.10/32
```

2. Enable RSVP, MPLS, and OSPF on the interfaces.

```
[edit protocols]
user@PE1# set rsvp interface fe-2/0/10.1
user@PE1# set rsvp interface fe-2/0/9.8
user@PE1# set rsvp interface fe-2/0/8.9
user@PE1# set rsvp interface lo0.1
user@PE1# set mpls interface fe-2/0/10.1
user@PE1# set mpls interface fe-2/0/9.8
user@PE1# set mpls interface fe-2/0/8.9
user@PE1# set mpls interface lo0.1
user@PE1# set ospf area 0.0.0.0 interface ge-2/0/2.0
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/10.1
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/9.8
user@PE1# set ospf area 0.0.0.0 interface fe-2/0/8.9
user@PE1# set ospf area 0.0.0.0 interface lo0.1
```

3. Configure the MPLS point-to-multipoint LSPs.

```
[edit protocols]
user@PE1# set mpls label-switched-path PE1-PE2 to 100.50.50.50
user@PE1# set mpls label-switched-path PE1-PE2 p2mp p2mp1
user@PE1# set mpls label-switched-path PE1-PE3 to 100.70.70.70
user@PE1# set mpls label-switched-path PE1-PE3 p2mp p2mp1
user@PE1# set mpls label-switched-path PE1-PE4 to 100.40.40.40
user@PE1# set mpls label-switched-path PE1-PE4 p2mp p2mp1
```

4. (Optional) Enable link protection on the LSPs.

   Link protection helps to ensure that traffic sent over a specific interface to a neighboring router can continue to reach the router if that interface fails.

```
[edit protocols]
user@PE1# set mpls label-switched-path PE1-PE2 link-protection
user@PE1# set mpls label-switched-path PE1-PE3 link-protection
user@PE1# set mpls label-switched-path PE1-PE4 link-protection
```

5. Enable MPLS to perform traffic engineering for OSPF.

```
[edit protocols]
user@PE1# set mpls traffic-engineering bgp-igp
```

   This causes the ingress routes to be installed in the **inet.0** routing table. By default, MPLS performs traffic engineering for BGP only. You need to enable MPLS traffic engineering on the ingress LSR only.

6. Enable traffic engineering for OSPF.

```
[edit protocols]
user@PE1# set ospf traffic-engineering
```

   This causes the shortest-path first (SPF) algorithm to take into account the LSPs configured under MPLS.

7. Configure the router ID.

```
[edit routing-options]
user@PE1# set router-id 100.10.10.10
```

 Copyright © 2012, Juniper Networks, Inc.

8. Configure static IP unicast routes with the point-to-multipoint LSP name as the next hop for each route.

```
[edit routing-options]
user@PE1# set static route 5.5.5.0/24 p2mp-lsp-next-hop p2mp1
user@PE1# set static route 7.7.7.0/24 p2mp-lsp-next-hop p2mp1
user@PE1# set static route 4.4.4.0/24 p2mp-lsp-next-hop p2mp1
```

9. If you are done configuring the device, commit the configuration.

```
[edit]
user@PE1# commit
```

Configuring the Transit and Egress LSRs (Devices P2, P3, P4, PE2, PE3, and PE4)

Step-by-Step Procedure

To configure the transit and egress LSRs:

1. Configure the interfaces, interface encapsulation, and protocol families.

```
[edit]
user@P2# set interfaces fe-2/0/10 unit 2 description P2-to-PE1
user@P2# set interfaces fe-2/0/10 unit 2 family inet address 2.2.2.2/24
user@P2# set interfaces fe-2/0/10 unit 2 family mpls
user@P2# set interfaces fe-2/0/9 unit 10 description P2-to-PE2
user@P2# set interfaces fe-2/0/9 unit 10 family inet address 5.5.5.1/24
user@P2# set interfaces fe-2/0/9 unit 10 family mpls
user@P2# set interfaces lo0 unit 2 family inet address 100.20.20.20/32

user@PE2# set interfaces ge-2/0/3 unit 0 description PE2-to-CE2
user@PE2# set interfaces ge-2/0/3 unit 0 family inet address 10.0.224.10/30
user@PE2# set interfaces fe-2/0/10 unit 5 description PE2-to-P2
user@PE2# set interfaces fe-2/0/10 unit 5 family inet address 5.5.5.2/24
user@PE2# set interfaces fe-2/0/10 unit 5 family mpls
user@PE2# set interfaces lo0 unit 5 family inet address 100.50.50.50/32

user@P3# set interfaces fe-2/0/10 unit 6 description P3-to-PE1
user@P3# set interfaces fe-2/0/10 unit 6 family inet address 6.6.6.2/24
user@P3# set interfaces fe-2/0/10 unit 6 family mpls
user@P3# set interfaces fe-2/0/9 unit 11 description P3-to-PE3
user@P3# set interfaces fe-2/0/9 unit 11 family inet address 7.7.7.1/24
user@P3# set interfaces fe-2/0/9 unit 11 family mpls
user@P3# set interfaces lo0 unit 6 family inet address 100.60.60.60/32

user@PE3# set interfaces ge-2/0/1 unit 0 description PE3-to-CE3
user@PE3# set interfaces ge-2/0/1 unit 0 family inet address 10.0.134.10/30
user@PE3# set interfaces fe-2/0/10 unit 7 description PE3-to-P3
user@PE3# set interfaces fe-2/0/10 unit 7 family inet address 7.7.7.2/24
user@PE3# set interfaces fe-2/0/10 unit 7 family mpls
user@PE3# set interfaces lo0 unit 7 family inet address 100.70.70.70/32

user@P4# set interfaces fe-2/0/10 unit 3 description P4-to-PE1
user@P4# set interfaces fe-2/0/10 unit 3 family inet address 3.3.3.2/24
user@P4# set interfaces fe-2/0/10 unit 3 family mpls
user@P4# set interfaces fe-2/0/9 unit 12 description P4-to-PE4
user@P4# set interfaces fe-2/0/9 unit 12 family inet address 4.4.4.1/24
```

```
user@P4# set interfaces fe-2/0/9 unit 12 family mpls
user@P4# set interfaces lo0 unit 3 family inet address 100.30.30.30/32

user@PE4# set interfaces ge-2/0/0 unit 0 description PE4-to-CE4
user@PE4# set interfaces ge-2/0/0 unit 0 family inet address 10.0.104.9/30
user@PE4# set interfaces fe-2/0/10 unit 4 description PE4-to-P4
user@PE4# set interfaces fe-2/0/10 unit 4 family inet address 4.4.4.2/24
user@PE4# set interfaces fe-2/0/10 unit 4 family mpls
user@PE4# set interfaces lo0 unit 4 family inet address 100.40.40.40/32
```

2. Enable RSVP, MPLS, and OSPF on the interfaces.

```
[edit]
user@P2# set protocols rsvp interface fe-2/0/10.2
user@P2# set protocols rsvp interface fe-2/0/9.10
user@P2# set protocols rsvp interface lo0.2
user@P2# set protocols mpls interface fe-2/0/10.2
user@P2# set protocols mpls interface fe-2/0/9.10
user@P2# set protocols mpls interface lo0.2
user@P2# set protocols ospf area 0.0.0.0 interface fe-2/0/10.2
user@P2# set protocols ospf area 0.0.0.0 interface fe-2/0/9.10
user@P2# set protocols ospf area 0.0.0.0 interface lo0.2

user@PE2# set protocols rsvp interface fe-2/0/10.5
user@PE2# set protocols rsvp interface lo0.5
user@PE2# set protocols mpls interface fe-2/0/10.5
user@PE2# set protocols mpls interface lo0.5
user@PE2# set protocols ospf area 0.0.0.0 interface ge-2/0/3.0
user@PE2# set protocols ospf area 0.0.0.0 interface fe-2/0/10.5
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.5

user@P3# set protocols rsvp interface fe-2/0/10.6
user@P3# set protocols rsvp interface fe-2/0/9.11
user@P3# set protocols rsvp interface lo0.6
user@P3# set protocols mpls interface fe-2/0/10.6
user@P3# set protocols mpls interface fe-2/0/9.11
user@P3# set protocols mpls interface lo0.6
user@P3# set protocols ospf area 0.0.0.0 interface fe-2/0/10.6
user@P3# set protocols ospf area 0.0.0.0 interface fe-2/0/9.11
user@P3# set protocols ospf area 0.0.0.0 interface lo0.6

user@PE3# set protocols rsvp interface fe-2/0/10.7
user@PE3# set protocols rsvp interface lo0.7
user@PE3# set protocols mpls interface fe-2/0/10.7
user@PE3# set protocols mpls interface lo0.7
user@PE3# set protocols ospf area 0.0.0.0 interface ge-2/0/1.0
user@PE3# set protocols ospf area 0.0.0.0 interface fe-2/0/10.7
user@PE3# set protocols ospf area 0.0.0.0 interface lo0.7

user@P4# set protocols rsvp interface fe-2/0/10.3
user@P4# set protocols rsvp interface fe-2/0/9.12
user@P4# set protocols rsvp interface lo0.3
user@P4# set protocols mpls interface fe-2/0/10.3
user@P4# set protocols mpls interface fe-2/0/9.12
```

 Copyright © 2012, Juniper Networks, Inc.

```
user@P4# set protocols mpls interface lo0.3
user@P4# set protocols ospf area 0.0.0.0 interface fe-2/0/10.3
user@P4# set protocols ospf area 0.0.0.0 interface fe-2/0/9.12
user@P4# set protocols ospf area 0.0.0.0 interface lo0.3

user@PE4# set protocols rsvp interface fe-2/0/10.4
user@PE4# set protocols rsvp interface lo0.4
user@PE4# set protocols mpls interface fe-2/0/10.4
user@PE4# set protocols mpls interface lo0.4
user@PE4# set protocols ospf area 0.0.0.0 interface ge-2/0/0.0
user@PE4# set protocols ospf area 0.0.0.0 interface fe-2/0/10.4
user@PE4# set protocols ospf area 0.0.0.0 interface lo0.4
```

3. Enable traffic engineering for OSPF.

```
[edit]
user@P2# set protocols ospf traffic-engineering

user@P2# set protocols ospf traffic-engineering

user@P3# set protocols ospf traffic-engineering

user@PE2# set protocols ospf traffic-engineering

user@PE3# set protocols ospf traffic-engineering

user@PE4# set protocols ospf traffic-engineering
```

This causes the shortest-path first (SPF) algorithm to take into account the LSPs configured under MPLS.

4. Configure the router IDs.

```
[edit]
user@P2# set routing-options router-id 100.20.20.20

user@P3# set routing-options router-id 100.60.60.60

user@P4# set routing-options router-id 100.30.30.30

user@PE2# set routing-options router-id 100.50.50.50

user@PE3# set routing-options router-id 100.70.70.70

user@PE4# set routing-options router-id 100.40.40.40
```

5. If you are done configuring the devices, commit the configuration.

```
[edit]
user@host# commit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device PE1

```
user@PE1# show interfaces
ge-2/0/2 {
   unit 0 {
      description R1-to-CE1;
      family inet {
         address 10.0.244.10/30;
      }
   }
}
fe-2/0/10 {
   unit 1 {
      description PE1-to-P2;
      family inet {
         address 2.2.2.1/24;
      }
      family mpls;
   }
}
fe-2/0/9 {
   unit 8 {
      description PE1-to-P2;
      family inet {
         address 6.6.6.1/24;
      }
      family mpls;
   }
}
fe-2/0/8 {
   unit 9 {
      description PE1-to-P3;
      family inet {
         address 3.3.3.1/24;
      }
      family mpls;
   }
}
lo0 {
   unit 1 {
      family inet {
         address 100.10.10.10/32;
      }
   }
}

user@PE1# show protocols
rsvp {
   interface fe-2/0/10.1;
   interface fe-2/0/9.8;
   interface fe-2/0/8.9;
   interface lo0.1;
}
```

 Copyright © 2012, Juniper Networks, Inc.

```
mpls {
   traffic-engineering bgp-igp;
   label-switched-path PE1-to-PE2 {
      to 100.50.50.50;
      link-protection;
      p2mp p2mp1;
   }
   label-switched-path PE1-to-PE3 {
      to 100.70.70.70;
      link-protection;
      p2mp p2mp1;
   }
   label-switched-path PE1-to-PE4 {
      to 100.40.40.40;
      link-protection;
      p2mp p2mp1;
   }
   interface fe-2/0/10.1;
   interface fe-2/0/9.8;
   interface fe-2/0/8.9;
   interface lo0.1;
}
ospf {
   traffic-engineering;
   area 0.0.0.0 {
      interface ge-2/0/2.0;
      interface fe-2/0/10.1;
      interface fe-2/0/9.8;
      interface fe-2/0/8.9;
      interface lo0.1;
   }
}
```

```
user@PE1# show routing-options
static {
   route 5.5.5.0/24 {
      p2mp-lsp-next-hop p2mp1;
   }
   route 7.7.7.0/24 {
      p2mp-lsp-next-hop p2mp1;
   }
   route 4.4.4.0/24 {
      p2mp-lsp-next-hop p2mp1;
   }
}
router-id 100.10.10.10;
```

Device P2

```
user@P2# show interfaces
fe-2/0/10 {
   unit 2 {
      description P2-to-PE1;
      family inet {
         address 2.2.2.2/24;
      }
      family mpls;
   }
fe-2/0/9 {
```

```
    unit 10 {
      description P2-to-PE2;
      family inet {
        address 5.5.5.1/24;
      }
      family mpls;
    }
  }
  lo0 {
    unit 2 {
      family inet {
        address 100.20.20.20/32;
      }
    }
  }

  user@P2# show protocols
  rsvp {
    interface fe-2/0/10.2;
    interface fe-2/0/9.10;
    interface lo0.2;
  }
  mpls {
    interface fe-2/0/10.2;
    interface fe-2/0/9.10;
    interface lo0.2;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface fe-2/0/10.2;
      interface fe-2/0/9.10;
      interface lo0.2;
    }
  }

  user@P2# show routing-options
  router-id 100.20.20.20;
```

Device P3

```
  user@P3# show interfaces
  fe-2/0/10 {
    unit 6 {
      description P3-to-PE1;
      family inet {
        address 6.6.6.2/24;
      }
      family mpls;
    }
  }
  fe-2/0/9 {
    unit 11 {
      description P3-to-PE3;
      family inet {
        address 7.7.7.1/24;
      }
      family mpls;
```


Copyright © 2012, Juniper Networks, Inc.

```
    }
  }
  lo0 {
    unit 6 {
      family inet {
        address 100.60.60.60/32;
      }
    }
  }

  user@P3# show protocols
  rsvp {
    interface fe-2/0/10.6;
    interface fe-2/0/9.11;
    interface lo0.6;
  }
  mpls {
    interface fe-2/0/10.6;
    interface fe-2/0/9.11;
    interface lo0.6;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface fe-2/0/10.6;
      interface fe-2/0/9.11;
      interface lo0.6;
    }
  }

  user@P2# show routing-options
  router-id 100.60.60.60;
```

Device P4

```
  user@P4# show interfaces
  fe-2/0/10 {
    unit 3 {
      description P4-to-PE1;
      family inet {
        address 3.3.3.2/24;
      }
      family mpls;
    }
  }
  fe-2/0/9 {
    unit 12 {
      description P4-to-PE4;
      family inet {
        address 4.4.4.1/24;
      }
      family mpls;
    }
  }
  lo0 {
    unit 3 {
      family inet {
        address 100.30.30.30/32;
```

```
        }
    }
}

user@P4# show protocols
rsvp {
    interface fe-2/0/10.3;
    interface fe-2/0/9.12;
    interface lo0.3;
}
mpls {
    interface fe-2/0/10.3;
    interface fe-2/0/9.12;
    interface lo0.3;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface fe-2/0/10.3;
        interface fe-2/0/9.12;
        interface lo0.3;
    }
}

user@P3# show routing-options
router-id 100.30.30.30;
```

Device PE2

```
user@PE2# show interfaces
    ge-2/0/3 {
        unit 0 {
            description PE2-to-CE2;
            family inet {
                address 10.0.224.10/30;
            }
        }
    }
    fe-2/0/10 {
        unit 5 {
            description PE2-to-P2;
            family inet {
                address 5.5.5.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 5 {
            family inet {
                address 100.50.50.50/32;
            }
        }
    }
}

user@PE2# show protocols
rsvp {
```


Copyright © 2012, Juniper Networks, Inc.

```
    interface fe-2/0/10.5;
    interface lo0.5;
  }
  mpls {
    interface fe-2/0/10.5;
    interface lo0.5;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface ge-2/0/3.0;
      interface fe-2/0/10.5;
      interface lo0.5;
    }
  }

  user@PE2# show routing-options
  router-id 100.50.50.50;
```

Device PE3

```
  user@PE3# show interfaces
    ge-2/0/1 {
      unit 0 {
        description PE3-to-CE3;
        family inet {
          address 10.0.134.10/30;
        }
      }
    }
    fe-2/0/10 {
      unit 7 {
        description PE3-to-P3;
        family inet {
          address 7.7.7.2/24;
        }
        family mpls;
      }
    }
    lo0 {
      unit 7 {
        family inet {
          address 100.70.70.70/32;
        }
      }
    }
  }

  user@PE3# show protocols
  rsvp {
    interface fe-2/0/10.7;
    interface lo0.7;
  }
  mpls {
    interface fe-2/0/10.7;
    interface lo0.7;
  }
  ospf {
```

```
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/1.0;
        interface fe-2/0/10.7;
        interface lo0.7;
    }
}
```

```
user@PE3# show routing-options
router-id 100.70.70.70;
```

Device PE4

```
user@PE4# show interfaces
    ge-2/0/0 {
        unit 0 {
            description PE4-to-CE4;
            family inet {
                address 10.0.104.9/30;
            }
        }
    }
    fe-2/0/10 {
        unit 4 {
            description PE4-to-P4;
            family inet {
                address 4.4.4.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 4 {
            family inet {
                address 100.40.40.40/32;
            }
        }
    }
}
```

```
user@PE4# show protocols
rsvp {
    interface fe-2/0/10.4;
    interface lo0.4;
}
mpls {
    interface fe-2/0/10.4;
    interface lo0.4;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-2/0/0.0;
        interface fe-2/0/10.4;
        interface lo0.4;
    }
}
```

```
user@PE4# show routing-options
```


Copyright © 2012, Juniper Networks, Inc.

```
router-id 100.40.40.40;
```

Configuring Device CE1

Step-by-Step Procedure

To configure Device CE1:

1. Configure an interface to Device PE1.

```
[edit interfaces]
user@CE1# set ge-1/3/2 unit 0 family inet address 10.0.244.9/30
user@CE1# set ge-1/3/2 unit 0 description CE1-to-PE1
```

2. Configure static routes from Device CE1 to the three other customer networks, with Device PE1 as the next hop.

```
[edit routing-options]
user@CE1# set static route 10.0.104.8/30 next-hop 10.0.244.10
user@CE1# set static route 10.0.134.8/30 next-hop 10.0.244.10
user@CE1# set static route 10.0.224.8/30 next-hop 10.0.244.10
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE1# commit
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE1# show interfaces
ge-1/3/2 {
   unit 0 {
      family inet {
         address 10.0.244.9/30;
         description CE1-to-PE1;
      }
   }
}

user@CE1# show routing-options
static {
   route 10.0.104.8/30 next-hop 10.0.244.10;
   route 10.0.134.8/30 next-hop 10.0.244.10;
   route 10.0.224.8/30 next-hop 10.0.244.10;
}
```

Configuring Device CE2

Step-by-Step Procedure

To configure Device CE2:

1. Configure an interface to Device PE2.

```
[edit interfaces]
user@CE2# set ge-1/3/3 unit 0 family inet address 10.0.224.9/30
user@CE2# set ge-1/3/3 unit 0 description CE2-to-PE2
```

2. Configure a static route from Device CE2 to CE1, with Device PE2 as the next hop.

```
[edit routing-options]
user@CE2# set static route 10.0.244.8/30 next-hop 10.0.224.10
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE2# commit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE2# show interfaces
ge-1/3/3 {
   unit 0 {
      family inet {
         address 10.0.224.9/30;
         description CE2-to-PE2;
      }
   }
}

user@CE2# show routing-options
static {
   route 10.0.244.8/30 next-hop 10.0.224.10;
}
```

Configuring Device CE3

Step-by-Step Procedure To configure Device CE3:

1. Configure an interface to Device PE3.

```
[edit interfaces]
user@CE3# set ge-2/0/1 unit 0 family inet address 10.0.134.9/30
user@CE3# set ge-2/0/1 unit 0 description CE3-to-PE3
```

2. Configure a static route from Device CE3 to CE1, with Device PE3 as the next hop.

```
[edit routing-options]
user@CE3# set static route 10.0.244.8/30 next-hop 10.0.134.10
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE3# commit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE3# show interfaces
ge-2/0/1 {
   unit 0 {
      family inet {
```

 Copyright © 2012, Juniper Networks, Inc.

```
            address 10.0.134.9/30;
            description CE3-to-PE3;
        }
    }
}
```

```
user@CE3# show routing-options
static {
    route 10.0.244.8/30 next-hop 10.0.134.10;
}
```

Configuring Device CE4

Step-by-Step Procedure

To configure Device CE4:

1. Configure an interface to Device PE4.

```
[edit interfaces]
user@CE4# set ge-3/1/3 unit 0 family inet address 10.0.104.10/30
user@CE4# set ge-3/1/3 unit 0 description CE4-to-PE4
```

2. Configure a static route from Device CE4 to CE1, with Device PE4 as the next hop.

```
[edit routing-options]
user@CE4# set static route 10.0.244.8/30 next-hop 10.0.104.9
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE4# commit
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE4# show interfaces
ge-3/1/3 {
    unit 0 {
        family inet {
            address 10.0.104.10/30;
            description CE4-to-PE4;
        }
    }
}
```

```
user@CE4# show routing-options
static {
    route 10.0.244.8/30 next-hop 10.0.104.9;
}
```

Verification

Confirm that the configuration is working properly.

- Verifying Connectivity on page 78
- Verifying the State of the Point-to-Multipoint LSP on page 78
- Checking the Forwarding Table on page 79

Verifying Connectivity

Purpose Make sure that the devices can ping each other.

Action Run the **ping** command from CE1 to the interface on CE2 connecting to PE2.

```
user@CE1> ping 10.0.224.9
PING 10.0.224.9 (10.0.224.9): 56 data bytes
64 bytes from 10.0.224.9: icmp_seq=0 ttl=61 time=1.387 ms
64 bytes from 10.0.224.9: icmp_seq=1 ttl=61 time=1.394 ms
64 bytes from 10.0.224.9: icmp_seq=2 ttl=61 time=1.506 ms
^C
--- 10.0.224.9 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.387/1.429/1.506/0.055 ms
```

Run the **ping** command from CE1 to the interface on CE3 connecting to PE3.

```
user@CE1> ping 10.0.134.9
PING 10.0.134.9 (10.0.134.9): 56 data bytes
64 bytes from 10.0.134.9: icmp_seq=0 ttl=61 time=1.068 ms
64 bytes from 10.0.134.9: icmp_seq=1 ttl=61 time=1.062 ms
64 bytes from 10.0.134.9: icmp_seq=2 ttl=61 time=1.053 ms
^C
--- 10.0.134.9 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.053/1.061/1.068/0.006 ms
```

Run the **ping** command from CE1 to the interface on CE4 connecting to PE4.

```
user@CE1> ping 10.0.104.10
PING 10.0.104.10 (10.0.104.10): 56 data bytes
64 bytes from 10.0.104.10: icmp_seq=0 ttl=61 time=1.079 ms
64 bytes from 10.0.104.10: icmp_seq=1 ttl=61 time=1.048 ms
64 bytes from 10.0.104.10: icmp_seq=2 ttl=61 time=1.070 ms
^C
--- 10.0.104.10 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.048/1.066/1.079/0.013 ms
```

Verifying the State of the Point-to-Multipoint LSP

Purpose Make sure that the ingress, transit, and egress LSRs are in the Up state.

Action Run the **show mpls lsp p2mp** command on all of the LSRs. Only the ingress LSR is shown here.

```
user@PE1> show mpls lsp p2mp
```

 Copyright © 2012, Juniper Networks, Inc.

```
Ingress LSP: 1 sessions
P2MP name: p2mp1, P2MP branch count: 3
To              From            State Rt P     ActivePath       LSPname
100.40.40.40    100.10.10.10    Up     0 *                      PE1-PE4
100.70.70.70    100.10.10.10    Up     0 *                      PE1-PE3
100.50.50.50    100.10.10.10    Up     0 *                      PE1-PE2
Total 3 displayed, Up 3, Down 0
...
```

Checking the Forwarding Table

Purpose Make sure that the routes are set up as expected by running the **show route forwarding-table** command. Only the routes to the remote customer networks are shown here.

Action

```
user@PE1> show route forwarding-table
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
...
10.0.104.8/30      user     0 3.3.3.2            ucst  1006     6 fe-2/0/8.9
10.0.134.8/30      user     0 6.6.6.2            ucst  1010     6 fe-2/0/9.8
10.0.224.8/30      user     0 2.2.2.2            ucst  1008     6 fe-2/0/10.1
...
```

Related Documentation

- Point-to-Multipoint LSPs Overview in the *Junos OS MPLS Applications Configuration Guide*

Static Routes and CLNs

- Understanding Static Routes for CLNS on page 79
- Example: Configuring Static Routes for CLNS on page 79

Understanding Static Routes for CLNS

The Connectionless Network Service (CLNS) is an ISO Layer 3 protocol that uses network service access point (NSAP) reachability information instead of IPv4 or IPv6 prefixes.

You can configure static routes to exchange CLNS routes within a CLNS island. A *CLNS island* is typically an IS-IS level 1 area that is part of a single IGP routing domain. An island can contain more than one area. CLNS islands can be connected by VPNs.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring Static Routes for CLNS on page 79

Example: Configuring Static Routes for CLNS

This example shows how to configure static routes for CLNS.

- Requirements on page 80
- Overview on page 80

- Configuration on page 80
- Verification on page 81

Requirements

Before you begin, configure the network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

Overview

In this example, you configure static routes for CLNS. In the absence of an interior gateway protocol (IGP) on a certain link, a routing device might need to be configured with static routes for CLNS prefixes to be reachable by way of that link. This might be useful, for example, at an autonomous system (AS) boundary.

When you configure static routes for CLNS, consider the following tasks:

- Specify the **iso.0** routing table option to configure a primary instance CLNS static route.
- Specify the **instance-name.iso.0** routing table option to configure a CLNS static route for a particular routing instance.
- Specify the **route** ***nsap-prefix*** statement to configure the destination for the CLNS static route.
- Specify the **next-hop (*interface-name* | *iso-net*)** statement to configure the next hop, specified as an ISO network entity title (NET) or interface name.
- Include the **qualified-next-hop (*interface-name* | *iso-net*)** statement to configure a secondary backup next hop, specified as an ISO network entity title or interface name.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set routing-options rib iso.0 static iso-route 47.0005.80ff.f800.0000.ffff.ffff/152 next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4212
set routing-options rib iso.0 static iso-route
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152 next-hop t1-0/2/2.0
set routing-options rib iso.0 static iso-route 47.0005.80ff.f800.0000.eee0/152
  qualified-next-hop 47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 preference
  20
set routing-options rib iso.0 static iso-route 47.0005.80ff.f800.0000.eee0/152
  qualified-next-hop 47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 metric 10
```

Step-by-Step Procedure

To configure static routes for CLNS:

1. Configure the routes.

```
[edit routing-options rib iso.0 static]
user@host# set iso-route 47.0005.80ff.f800.0000.ffff.ffff/152 next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4212
```


Copyright © 2012, Juniper Networks, Inc.

```
user@host# set iso-route 47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152
  next-hop t1-0/2/2.0
user@host# set iso-route 47.0005.80ff.f800.0000.eee0/152 qualified-next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 preference 20
user@host# set iso-route 47.0005.80ff.f800.0000.eee0/152 qualified-next-hop
  47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 metric 10
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results Confirm your configuration by issuing the **show routing-options** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show routing-options
rib iso.0 {
  static {
    iso-route 47.0005.80ff.f800.0000.ffff.ffff/152 next-hop
      47.0005.80ff.f800.0000.0108.0001.1921.6800.4212;
    iso-route 47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152 next-hop t1-0/2/2.0;
    iso-route 47.0005.80ff.f800.0000.eee0/152 {
      qualified-next-hop 47.0005.80ff.f800.0000.0108.0001.1921.6800.4002 {
        preference 20;
        metric 10;
      }
    }
  }
}
```

Verification

Confirm that the configuration is working properly.

Checking the Routing Table

Purpose Make sure that the expected routes appear in the routing table.

Action

```
user@host> show route table iso.0

iso.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

47.0005.80ff.f800.0000.0108.0001.1921.6800.4212/152
                   *[Static/5] 00:00:25
                    > via t1-0/2/2.0
47.0005.80ff.f800.0000.eee0/84
                   *[Static/20] 00:04:01, metric 10, metric2 10
                    > to #75 0.12.0.34.0.56 via fe-0/0/1.0
47.0005.80ff.f800.0000.ffff.ffff/104
                   *[Static/5] 00:04:01, metric2 0
                    > via t1-0/2/2.0
```

Meaning The static routes appear in the routing table.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- *Junos OS Routing Protocols Configuration Guide*
- CLNS Configuration Overview
- Understanding Static Routes for CLNS on page 79

Default Static Route Behavior

- Understanding Static Routing Default Properties on page 82
- Example: Defining Default Behavior for All Static Routes on page 83

Understanding Static Routing Default Properties

The basic configuration of static routes defines properties for a particular route. To define a set of properties to be used as defaults on all static routes, set those properties as default values. For example:

```
[edit routing-options static]
defaults {
   retain;
   no-readvertise;
   passive;
}
route 0.0.0.0/0 next-hop 192.168.1.1;
route 192.168.47.5/32 {
   next-hop 10.10.10.10;
   qualified-next-hop 10.10.10.7 {
      preference 6;
   }
   preference 2;
}
```

In this example, the **retain**, **no-readvertise**, and **passive** attributes are set as defaults for all static routes. If any local setting for a particular route conflicts with the default values, the local setting supersedes the default.

Attributes that define static route behavior can be configured either at the individual route level or as a default behavior that applies to all static routes. In the case of conflicting configuration, the configuration at the individual route level overrides static route defaults. Using the conflicting statements is useful if you configure default values, because you can override the default behaviors by including the conflicting version of the statement at a more specific hierarchy level. For example, include the **readvertise** statement when configuring an individual route in the **route** portion of the **static** statement to override a **no-readvertise** option specified in the **defaults** portion of the statement. Include the **no-retain** statement when configuring an individual route in the **route** portion of the **static** statement to override a **retain** option specified in the **defaults** portion of the statement. Include the **active** statement when configuring an individual route in the **route** portion of the **static** statement to override a **passive** option specified in the **defaults** portion of the statement.

 Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Defining Default Behavior for All Static Routes on page 83

Example: Defining Default Behavior for All Static Routes

This example shows how to define the default behavior for all static routes and how to override the default behavior for specific static routes.

- Requirements on page 83
- Overview on page 83
- Configuration on page 83
- Verification on page 87

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

This example shows how to perform the following actions:

- Configure attributes that define static route behavior as a default behavior that applies to all static routes. The configured default behavior specify that static routes are not to be readvertised. (The Junos OS default behavior is to make static routes eligible to be readvertised.)
- Configure specific static routes that override the configured default behavior.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Router A

```
set interfaces fe-1/2/0 unit 4 description A->B
set interfaces fe-1/2/0 unit 4 family inet address 10.0.2.2/30
set protocols ospf area 0.0.0.0 interface fe-1/2/0.4
```

Router B

```
set interfaces fe-1/2/0 unit 3 description B->A
set interfaces fe-1/2/0 unit 3 family inet address 10.0.2.1/30
set interfaces fe-1/2/1 unit 6 description B->C
set interfaces fe-1/2/1 unit 6 family inet address 10.0.3.1/30
set protocols bgp group ext type external
set protocols bgp group ext peer-as 23
set protocols bgp group ext neighbor 10.0.3.2
set protocols ospf export send-static
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set policy-options policy-statement send-static from protocol static
set policy-options policy-statement send-static then accept
set routing-options static defaults no-readvertise
set routing-options static route 0.0.0.0/0 next-hop 10.0.3.2
```

Copyright © 2012, Juniper Networks, Inc.

```
set routing-options static route 0.0.0.0/0 readvertise
set routing-options static route 192.168.0.0/24 next-hop 10.0.3.2
set routing-options static route 192.168.0.0/24 readvertise
set routing-options static route 1.1.1.0/24 discard
set routing-options static route 2.2.2.0/24 discard
set routing-options static route 3.3.3.0/24 discard
set routing-options autonomous-system 17
```

Router C

```
set interfaces fe-1/2/0 unit 7 description B->C
set interfaces fe-1/2/0 unit 7 family inet address 10.0.3.2/30
set interfaces lo0 unit 5 family inet address 192.168.0.1/32
set protocols bgp group ext type external
set protocols bgp group ext peer-as 17
set protocols bgp group ext neighbor 10.0.3.1
set routing-options autonomous-system 23
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Router A:

1. Configure the interface to Router B.

```
[edit interfaces fe-1/2/0 unit 4]
user@A# set description A->B
user@A# set family inet address 10.0.2.2/30
```

2. Configure OSPF to form an OSPF peer relationship with Router B.

```
[edit protocols ospf area 0.0.0.0]
user@A# set interface fe-1/2/0.4
```

Step-by-Step Procedure

To configure Router B:

1. Configure the interfaces to Router A and Router C.

```
[edit interfaces]
user@B# set fe-1/2/0 unit 3 description B->A
user@B# set fe-1/2/0 unit 3 family inet address 10.0.2.1/30
user@B# set fe-1/2/1 unit 6 description B->C
user@B# set fe-1/2/1 unit 6 family inet address 10.0.3.1/30
```

2. Configure some static routes and the autonomous system (AS) number.

```
[edit routing-options static]
user@B# set route 0.0.0.0/0 next-hop 10.0.3.2
user@B# set route 192.168.0.0/24 next-hop 10.0.3.2
user@B# set route 1.1.1.0/24 discard
user@B# set route 2.2.2.0/24 discard
user@B# set route 3.3.3.0/24 discard
```

3. Configure the default behavior for static routes.

```
[edit routing-options static]
user@B# set defaults no-readvertise
```

4. For a subset of the static routes, override the default behavior.

Copyright © 2012, Juniper Networks, Inc.

```
[edit routing-options static]
user@B# set route 0.0.0.0/0 readvertise
user@B# set route 192.168.0.0/24 readvertise
```

5. Configure the routing policy.

   This policy exports static routes from the routing table into OSPF.

```
[edit policy-options policy-statement send-static]
user@B# set from protocol static
user@B# set then accept
```

6. Configure the routing protocols.

   The BGP configuration forms an external BGP (EBGP) peer relationship with Router C.

   The OSPF configuration forms an OSPF peer relationship with Router A and applies the **send-static** routing policy.

```
[edit protocols]
user@B# set bgp group ext type external
user@B# set bgp group ext peer-as 23
user@B# set bgp group ext neighbor 10.0.3.2
user@B# set ospf export send-static
user@B# set ospf area 0.0.0.0 interface fe-1/2/0.3
```

Step-by-Step Procedure

To configure Router C:

1. Create the interface to Router B, and configure loopback interface.

```
[edit interfaces ]
user@C# set fe-1/2/0 unit 7 description B->C
user@C# set fe-1/2/0 unit 7 family inet address 10.0.3.2/30
user@C# set lo0 unit 5 family inet address 192.168.0.1/32
```

2. Configure the EBGP peering session with Router B.

```
[edit protocols bgp group ext]
user@C# set type external
user@C# set peer-as 17
user@C# set neighbor 10.0.3.1
```

3. Configure the AS number.

```
[edit routing-options]
user@C# set autonomous-system 23
```

Results

Confirm your configuration by issuing the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands.

Router A

```
user@A# show interfaces
fe-1/2/0 {
   unit 4 {
      description A->B;
      family inet {
         address 10.0.2.2/30;
      }
   }
```

```
}

user@A# show protocols
ospf {
   area 0.0.0.0 {
      interface fe-1/2/0.4;
   }
}
```

Router B

```
user@B# show interfaces
interfaces {
   fe-1/2/0 {
      unit 3 {
         description B->A;
         family inet {
            address 10.0.2.1/30;
         }
      }
   }
   fe-1/2/1 {
      unit 6 {
         description B->C;
         family inet {
            address 10.0.3.1/30;
         }
      }
   }
}

user@B# show policy-options
policy-statement send-static {
   from protocol static;
   then accept;
}

user@B# show protocols
bgp {
   group ext {
      type external;
      peer-as 23;
      neighbor 10.0.3.2;
   }
}
ospf {
   export send-static;
   area 0.0.0.0 {
      interface fe-1/2/0.3;
   }
}

user@B# show routing-options
static {
   defaults {
      no-readvertise;
   }
   route 0.0.0.0/0 {
      next-hop 10.0.3.2;
```


Copyright © 2012, Juniper Networks, Inc.

```
        readvertise;
    }
    route 192.168.0.0/24 {
        next-hop 10.0.3.2;
        readvertise;
    }
    route 1.1.1.0/24 discard;
    route 2.2.2.0/24 discard;
    route 3.3.3.0/24 discard;
}
autonomous-system 17;
```

Router C

```
user@C# show interfaces
fe-1/2/0 {
    unit 7 {
        description B->C;
        family inet {
            address 10.0.3.2/30;
        }
    }
}
lo0 {
    unit 5 {
        family inet {
            address 192.168.0.1/32;
        }
    }
}

user@C# show protocols
bgp {
    group ext {
        type external;
        peer-as 17;
        neighbor 10.0.3.1;
    }
}

user@C# show routing-options
autonomous-system 23;
```

If you are done configuring the devices, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Checking the Routing Tables

Purpose Make sure that the static routes appear (or do not appear) as expected in the routing tables.

Action

1. On Router B, run the **show route protocol static** command to make sure that all of the static routes appear in Router B's routing table.

```
user@B> show route protocols static
```

```
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:52:43
                    > to 10.0.3.2 via ft-1/2/1.6
1.1.1.0/24         *[Static/5] 00:45:12
                      Discard
2.2.2.0/24         *[Static/5] 00:45:12
                      Discard
3.3.3.0/24         *[Static/5] 00:45:12
                      Discard
192.168.0.0/24     *[Static/5] 00:52:43
                    > to 10.0.3.2 via ft-1/2/1.6
```

2. On Router A, run the **show route protocol ospf** command to make sure that the 1.1.1.1/24, 2.2.2.2/24, and 3.3.3.0/24 routes do not appear in Router A's routing table, and that the routes 0.0.0.0/0 and 192.168.0.0/24 do appear as routes learned through an external source. OSPF external routes have a reference of 150.

```
user@A> show route protocols ospf

inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[OSPF/150] 00:44:47, metric 0, tag 0
                    > to 10.0.2.1 via ft-1/2/0.4
192.168.0.0/24     *[OSPF/150] 00:46:01, metric 0, tag 0
                    > to 10.0.2.1 via ft-1/2/0.4
224.0.0.5/32       *[OSPF/10] 00:56:21, metric 1
                      MultiRecv
```

3. On Router B, deactivate the **defaults no-readvertise** statement.

```
user@B# deactivate routing-options static defaults no-readvertise
```

4. On Router A, rerun the **show route protocol ospf** command to make sure that the previously unadvertised routes are now advertised.

```
user@A> show route protocols ospf


inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[OSPF/150] 00:49:24, metric 0, tag 0
                    > to 10.0.2.1 via lt-1/2/0.4
1.1.1.0/24         *[OSPF/150] 00:00:06, metric 0, tag 0
                    > to 10.0.2.1 via lt-1/2/0.4
2.2.2.0/24         *[OSPF/150] 00:00:06, metric 0, tag 0
                    > to 10.0.2.1 via lt-1/2/0.4
3.3.3.0/24         *[OSPF/150] 00:00:06, metric 0, tag 0
                    > to 10.0.2.1 via lt-1/2/0.4
192.168.0.0/24     *[OSPF/150] 00:50:38, metric 0, tag 0
                    > to 10.0.2.1 via lt-1/2/0.4
224.0.0.5/32       *[OSPF/10] 01:00:58, metric 1
                      MultiRecv
```

Meaning The **no-readvertise** configured default behavior is working as expected, as is the overriding **readvertise** statement.

 Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Static Routing Default Properties on page 82
- **show route terse** in the *Junos OS CLI Reference*
- Verifying the Static Route Configuration on page 89

Verifying the Static Route Configuration

Purpose Verify that the static routes are in the routing table and that those routes are active.

Action From the CLI, enter the **show route terse** command.

Sample Output

```
user@host> show route terse
inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* 192.168.47.5/32        S   5                        Reject
* 172.16.0.0/12      S   5                       >192.168.71.254
* 192.168.0.0/18     S   5                       >192.168.71.254
* 192.168.40.0/22    S   5                       >192.168.71.254
* 192.168.64.0/18    S   5                       >192.168.71.254
* 192.168.64.0/21    D   0                       >fxp0.0
* 192.168.71.246/32  L   0                        Local
* 192.168.220.4/30   D   0                       >ge-0/0/1.0
* 192.168.220.5/32   L   0                        Local
* 192.168.220.8/30   D   0                       >ge-0/0/2.0
* 192.168.220.9/32   L   0                        Local
* 192.168.220.12/30  D   0                       >ge-0/0/3.0
* 192.168.220.13/32  L   0                        Local
* 192.168.220.17/32  L   0                        Reject
* 192.168.220.21/32  L   0                        Reject
* 192.168.220.24/30  D   0                       >at-1/0/0.0
* 192.168.220.25/32  L   0                        Local
* 192.168.220.28/30  D   0                       >at-1/0/1.0
* 192.168.220.29/32  L   0                        Local
* 224.0.0.9/32       R 100          1             MultiRecv
```

Meaning The output shows a list of the routes that are currently in the **inet.0** routing table. Verify the following information:

- Each configured static route is present. Routes are listed in ascending order by IP address. Static routes are identified with an **S** in the protocol (**P**) column of the output.
- Each static route is active. Routes that are active show the next-hop IP address in the **Next hop** column. If a route's next-hop address is unreachable, the next-hop address is identified as **Reject**. These routes are not active routes, but they appear in the routing table because the **passive** attribute is set.
- The preference for each static route is correct. The preference for a particular route is listed in the **Prf** column of the output.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- show route terse in the *Junos OS Routing Protocols and Policies Command Reference*

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 5

Aggregate and Generated Routes

- Understanding Aggregate Routes on page 91
- Example: Summarizing Routes Through Route Aggregation on page 98

Understanding Aggregate Routes

Route aggregation allows you to combine groups of routes with common addresses into a single entry in the routing table. This decreases the size of the routing table as well as the number of route advertisements sent by the routing device.

An aggregate route becomes active when it has one or more *contributing routes*. A contributing route is an active route that is a more specific match for the aggregate destination. For example, for the aggregate destination **128.100.0.0/16**, routes to **128.100.192.0/19** and **128.100.67.0/24** are contributing routes, but routes to **128.0.0.0./8** and **128.0.0.0/16** are not.

A route can contribute only to a single aggregate route. However, an active aggregate route can recursively contribute to a less specific matching aggregate route. For example, an aggregate route to the destination **128.100.0.0/16** can contribute to an aggregate route to **128.96.0.0/13**.

When an aggregate route becomes active, it is installed in the routing table with the following information:

- Reject next hop—If a more-specific packet does not match a more-specific route, the packet is rejected and an ICMP unreachable message is sent to the packet's originator.
- Metric value as configured with the **aggregate** statement.
- Preference value that results from the policy filter on the primary contributor, if a filter is specified.
- AS path as configured in the **aggregate** statement, if any. Otherwise, the path is computed by aggregating the paths of all contributing routes.
- Community as configured in the **aggregate** statement, if any is specified.

NOTE: **You can configure only one aggregate route for each destination prefix.**

To configure aggregate routes in the default routing table (**inet.0**), include the **aggregate** statement:

```
aggregate {
  defaults {
    ... aggregate-options ...
  }
  route destination-prefix {
    policy policy-name;
    ... aggregate-options ...
  }
}
```

To configure aggregate routes in one of the other routing tables, or to explicitly configure aggregate routes in the default routing table (**inet.0**), include the **aggregate** statement:

```
rib routing-table-name {
  aggregate {
    defaults {
      ... aggregate-options ...
    }
    route destination-prefix {
      policy policy-name;
      ... aggregate-options ...
    }
  }
}
```

NOTE: You cannot configure aggregate routes for the IPv4 multicast routing table (inet.1) nor the IPv6 multicast routing table (inet6.1).

The **aggregate** statement consists of two parts:

- **defaults**—(Optional) Here you specify global aggregate route options. These are treated as global defaults and apply to all the aggregate routes you configure in the **aggregate** statement.
- **route**—Here you configure individual aggregate routes. In this part of the **aggregate** statement, you optionally can configure aggregate route options. These options apply to the individual destination only and override any options you configured in the **defaults** part of the **aggregate** statement.

When you configure an individual aggregate route in the **route** part of the **aggregate** statement, specify the destination of the route (in **route *destination-prefix***) in one of the following ways:

- ***network/mask-length***, where ***network*** is the network portion of the IP address and ***mask-length*** is the destination prefix length.
- **default** if this is the default route to the destination. This is equivalent to specifying an IP address of **0.0.0.0/0**.

Copyright © 2012, Juniper Networks, Inc.

After you have configured aggregate routes, you can have a protocol advertise the routes by configuring a policy that is then exported by a routing protocol.

You can associate a routing policy when configuring an aggregate route's destination prefix in the **routes** part of the **aggregate** statement. Doing so provides the equivalent of an import routing policy filter for the destination prefix. That is, each potential contributor to an aggregate route, along with any aggregate options, is passed through the policy filter. The policy then can accept or reject the route as a contributor to the aggregate route and, if the contributor is accepted, the policy can modify the default preferences.

The following algorithm is used to compare two aggregate contributing routes in order to determine which one is the primary or preferred contributor:

1. Compare the protocol's **preferences** of the contributing routes. The lower the preference, the better the route. This is similar to the comparison that is done while determining the best route for the routing table.
2. Compare the protocol's **preferences2** of the contributing routes. The lower preference2 value is better. If only one route has **preferences2**, then this route is preferred.
3. The preference values are the same. Proceed with a numerical comparison of the prefix values.
   a. The primary contributor is the numerically smallest prefix value.
   b. If the two prefixes are numerically equal, the primary contributor is the route that has the smallest prefix length value.
4. At this point, the two routes are the same. The primary contributor does not change. An additional next hop is available for the existing primary contributor.

A rejected contributor still can contribute to a less specific aggregate route. If you do not specify a policy filter, all candidate routes contribute to an aggregate route.

To associate a routing policy with an aggregate route, include the **policy** statement when configuring the route:

```
aggregate (defaults | route) {
  policy policy-name;
}
```

In the **defaults** and **route** parts of the **aggregate** statement, you can specify ***aggregate-options***, which define additional information about aggregate routes that is included with the route when it is installed in the routing table. All aggregate options are optional. Aggregate options that you specify in the **defaults** part of the **aggregate** statement are treated as global defaults and apply to all the aggregate routes you configure in the **aggregate** statement. Aggregate options that you specify in the **route** part of the **aggregate** statement override any global aggregate options and apply to that destination only.

To configure aggregate route options, include one or more of them in the **defaults** or **route** part of the **aggregate** statement:

```
[edit]
```

```
routing-options {
  aggregate {
    (defaults | route) {
      (active | passive);
      as-path <as-path> <origin (egp | igp | incomplete)> <atomic-aggregate> <aggregator
        as-number in-address>;
      community [ community-ids ];
      discard;
      (brief | full);
      (metric | metric2 | metric3 | metric4) metric <type type>;
      (preference | preference2 | color | color2) preference <type type>;
      tag string;
    }
  }
}
```

Configuring a Metric Value for Aggregate Routes

You can specify up to four metric values, starting with **metric** (for the first metric value) and continuing with **metric2**, **metric3**, and **metric4** by including one or more of the following statements:

```
aggregate (defaults | route) {
  (metric | metric2 | metric3 | metric4) metric <type type>;
}
```

For a list of hierarchy levels at which you can include these statements, see the statement summary sections for these statements.

In the **type** option, you can specify the type of route.

Configuring a Preference Value for Aggregate Routes

By default, aggregate routes have a preference value of 130. If the routing table contains a dynamic route to a destination that has a better (lower) preference value than this, the dynamic route is chosen as the active route and is installed in the forwarding table.

To modify the default preference value, specify a primary preference value (**preference**). You also can specify secondary preference value (**preference2**); and colors, which are even finer-grained preference values (**color** and **color2**). To do this, include one or more of the following statements:

```
aggregate (defaults | route) {
  (preference | preference2 | color | color2) preference <type type>;
}
```

For a list of hierarchy levels at which you can include these statements, see the statement summary sections for these statements.

The preference value can be a number in the range from 0 through 4,294,967,295 ($2^{32} - 1$) with a lower number indicating a more preferred route. For more information about preference values, see Route Preferences Overview.

In the **type** option, you can specify the type of route.

Copyright © 2012, Juniper Networks, Inc.

Configuring the Next Hop for Aggregate Routes

By default, when aggregate routes are installed in the routing table, the next hop is configured as a reject route. That is, the packet is rejected and an ICMP unreachable message is sent to the packet's originator.

When you configure an individual route in the **route** part of the **aggregate** statement, or when you configure the defaults for aggregate routes, you can specify a discard next hop. This means that if a more specific packet does not match a more specific route, the packet is rejected and a reject route for this destination is installed in the routing table, but ICMP unreachable messages are not sent.

Being able to discard next hops allows you to originate a summary route, which can be advertised through dynamic routing protocols, and allows you to discard received traffic that does not match a more specific route than the summary route. To discard next hops, include the **discard** option:

```
discard;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Associating BGP Communities with Aggregate Routes

By default, no BGP community information is associated with aggregate routes. To associate community information with the routes, include the **community** option:

```
aggregate (defaults | route) {
   community [ community-ids ];
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement. ***community-value*** is the community identifier and can be a number in the range from 0 through 65,535.

community-ids is one or more community identifiers for either communities or extended communities.

The format for community identifiers is:

as-number:*community-value*

as-number is the AS number and can be a value in the range from 1 through 65,534.

You also can specify ***community-ids*** for communities as one of the following well-known community names, which are defined in RFC 1997:

- **no-export**—Routes containing this community name are not advertised outside a BGP confederation boundary.
- **no-advertise**—Routes containing this community name are not advertised to other BGP peers.
- **no-export-subconfed**—Routes containing this community name are not advertised to external BGP peers, including peers in other members' ASs inside a BGP confederation.

Copyright © 2012, Juniper Networks, Inc.

You can explicitly exclude BGP community information with an aggregate route using the **none** option. Include **none** when configuring an individual route in the **route** portion of the **aggregate** statement to override a **community** option specified in the **defaults** portion of the statement.

NOTE: Extended community attributes are not supported at the **[edit routing-options]** hierarchy level. You must configure extended communities at the **[edit policy-options]** hierarchy level. For information about configuring extended communities information, see the "Configuring the Extended Communities Attribute" section in the *Junos OS Policy Framework Configuration Guide*. For information about configuring 4-byte AS numbers and extended communities, see Configuring 4-Byte AS Numbers and BGP Extended Community Attributes in the *Using 4-Byte Autonomous System Numbers in BGP Networks Technology Overview*.

Associating AS Paths with Aggregate Routes

By default, the AS path for aggregate routes is built from the component routes. To manually specify the AS path and associate AS path information with the routes, include the **as-path** option:

```
aggregate (defaults | route) {
  as-path <as-path> <origin (egp | igp | incomplete)> <atomic-aggregate> <aggregator
    as-number in-address>;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

as-path is the AS path to include with the route. It can include a combination of individual AS path numbers and AS sets. Enclose sets in brackets ([ ]). The first AS number in the path represents the AS immediately adjacent to the local AS. Each subsequent number represents an AS that is progressively farther from the local AS, heading toward the origin of the path.

NOTE: In Junos OS Release 9.1 and later, the numeric AS range is extended to provide BGP support for 4-byte AS numbers, as defined in RFC 4893, *BGP Support for Four-octet AS Number Space*. For the AS number, you can configure a value from 1 through 4,294,967,295. All releases of Junos OS support 2-byte AS numbers. The 2-byte AS number range is 1 through 65,535 (this is a subset of the 4-byte range).

In Junos OS Release 9.2 and later, you can also configure a 4-byte AS number using the AS-dot notation format of two integer values joined by a period: *<16-bit high-order value in decimal>.<16-bit low-order value in decimal>*. For example, the 4-byte AS number of 65,546 in plain-number format is represented as 1.10 in the AS-dot notation format. You can specify a value in the range from 0.0 through 65535.65535 in AS-dot notation format.

Copyright © 2012, Juniper Networks, Inc.

You also can specify the AS path using the BGP origin attribute, which indicates the origin of the AS path information:

- **egp**—Path information originated in another AS.
- **igp**—Path information originated within the local AS.
- **incomplete**—Path information was learned by some other means.

To attach the BGP **ATOMIC_AGGREGATE** path attribute to the aggregate route, specify the **atomic-aggregate** option. This path attribute indicates that the local system selected a less specific route rather than a more specific route.

To attach the BGP **AGGREGATOR** path attribute to the aggregate route, specify the **aggregator** option. When using this option, you must specify the last AS number that formed the aggregate route (encoded as two octets), followed by the IP address of the BGP system that formed the aggregate route.

Including AS Numbers in Aggregate Route Paths

By default, all AS numbers from all contributing paths are included in the aggregate route's path. To include only the longest common leading sequences from the contributing AS paths, include the **brief** option when configuring the route. If doing this results in AS numbers being omitted from the aggregate route, the BGP **ATOMIC_ATTRIBUTE** path attribute is included with the aggregate route.

```
aggregate (defaults | route) {
   brief;
}
```

To explicitly have all AS numbers from all contributing paths be included in the aggregate route's path, include the **full** option when configuring routes. Include this option when configuring an individual route in the **route** portion of the **aggregate** statement to override a **retain** option specified in the **defaults** portion of the statement.

```
aggregate (defaults | route) {
   full;
}
```

For a list of hierarchy levels at which you can include these statements, see the statement summary sections for these statements.

Configuring an OSPF Tag String for Aggregate Routes

By default, no OSPF tag strings are associated with aggregate routes. You can specify an OSPF tag string by including the **tag** option:

```
aggregate (defaults | route) {
   tag string;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Copyright © 2012, Juniper Networks, Inc.

Controlling Retention of Inactive Aggregate Routes in the Routing and Forwarding Tables

Static routes are only removed from the routing table if the next hop becomes unreachable, which happens if there are no contributing routes. To have an aggregate route remain continually installed in the routing and forwarding tables, include the **passive** option when configuring the route:

```
aggregate (defaults | route) {
   passive;
}
```

Routes that have been configured to remain continually installed in the routing and forwarding tables are marked with **reject** next hops when they are inactive.

To explicitly remove aggregate routes when they become inactive, include the **active** option when configuring routes. Include this option when configuring an individual route in the **route** portion of the **aggregate** statement to override a **passive** option specified in the **defaults** portion of the statement.

```
aggregate (defaults | route) {
   active;
}
```

Related Documentation

- Example: Summarizing Routes Through Route Aggregation on page 98

Example: Summarizing Routes Through Route Aggregation

This example shows how to summarize routes by configuring aggregate routes.

- Requirements on page 98
- Overview on page 98
- Configuration on page 99
- Verification on page 103

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, Device R1 is connected to customer networks 10.200.1.0/24 and 10.200.2.0/24. For demonstration purposes, these routes are represented in this example as loopback interfaces on Device R1.

Device R2 has static routes configured to reach Device R1's customer networks. Device R2 also has a routing policy configured to advertise all static routes to its neighbors in autonomous system (AS) 65001.

Copyright © 2012, Juniper Networks, Inc.

Device R3 is in AS 65001 and receives the static routes from Device R2. When Device R3 sends information about these routes to Device ISP, the information is summarized as a single aggregate route. The aggregate route is 10.200.0.0/16.

Device ISP injects a default route into AS 65001, and Device R3 advertises the default route.

This example shows the configuration for all of the devices and the step-by-step configuration on Device R3.

Figure 14 on page 99 shows the sample network.

Figure 14: Aggregate Route Advertised to an ISP

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces ge-1/2/0 unit 2 description R1->R2
set interfaces ge-1/2/0 unit 2 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 10.200.1.1/32
set interfaces lo0 unit 1 family inet address 10.200.2.2/32
set protocols bgp group ext type external
set protocols bgp group ext peer-as 65001
set protocols bgp group ext neighbor 10.0.0.2
set protocols ospf area 0.0.0.0 interface ge-1/2/0.2
set routing-options autonomous-system 65003
```

Device R2

```
set interfaces ge-1/2/0 unit 1 description R2->R1
set interfaces ge-1/2/0 unit 1 family inet address 10.0.0.2/30
set interfaces ge-1/2/1 unit 4 description R2->R3
set interfaces ge-1/2/1 unit 4 family inet address 10.0.2.2/30
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
```

```
set protocols bgp group int type internal
set protocols bgp group int local-address 172.16.2.2
set protocols bgp group int export send-customer-routes
set protocols bgp group int neighbor 172.16.3.3
set protocols bgp group ext type external
set protocols bgp group ext peer-as 65003
set protocols bgp group ext neighbor 10.0.0.1
set protocols ospf area 0.0.0.0 interface ge-1/2/0.1
set protocols ospf area 0.0.0.0 interface ge-1/2/1.4
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set policy-options policy-statement send-customer-routes from protocol static
set policy-options policy-statement send-customer-routes then accept
set routing-options static route 10.200.1.0/24 next-hop 10.0.0.1
set routing-options static route 10.200.2.0/24 next-hop 10.0.0.1
set routing-options autonomous-system 65001
```

Device R3

```
set interfaces ge-1/2/0 unit 3 description R3->R2
set interfaces ge-1/2/0 unit 3 family inet address 10.0.2.1/30
set interfaces ge-1/2/1 unit 6 description R3->ISP
set interfaces ge-1/2/1 unit 6 family inet address 10.0.45.2/30
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols bgp group ext type external
set protocols bgp group ext export send-aggregate
set protocols bgp group ext peer-as 65000
set protocols bgp group ext neighbor 10.0.45.1
set protocols bgp group int type internal
set protocols bgp group int local-address 172.16.3.3
set protocols bgp group int neighbor 172.16.2.2
set protocols ospf export send-default
set protocols ospf area 0.0.0.0 interface ge-1/2/0.3
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set policy-options policy-statement send-aggregate term 1 from protocol aggregate
set policy-options policy-statement send-aggregate term 1 then accept
set policy-options policy-statement send-aggregate term suppress-specific-routes from
    route-filter 10.200.0.0/16 longer
set policy-options policy-statement send-aggregate term suppress-specific-routes then
    reject
set policy-options policy-statement send-default from route-filter 0.0.0.0/0 exact
set policy-options policy-statement send-default then accept
set routing-options aggregate route 10.200.0.0/16
set routing-options autonomous-system 65001
```

Device ISP

```
set interfaces ge-1/2/0 unit 7 family inet address 10.0.45.1/30
set protocols bgp group ext type external
set protocols bgp group ext export advertise-default
set protocols bgp group ext peer-as 65001
set protocols bgp group ext neighbor 10.0.45.2
set policy-options policy-statement advertise-default term 1 from route-filter 0.0.0.0/0
    exact
set policy-options policy-statement advertise-default term 1 then accept
set routing-options static route 0.0.0.0/0 discard
set routing-options autonomous-system 65000
```


Copyright © 2012, Juniper Networks, Inc.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device R3:

1. Configure the interfaces.

```
[edit interfaces]
user@R3# set ge-1/2/0 unit 3 description R3->R2
user@R3# set ge-1/2/0 unit 3 family inet address 10.0.2.1/30

user@R3# set ge-1/2/1 unit 6 description R3->ISP
user@R3# set ge-1/2/1 unit 6 family inet address 10.0.45.2/30

user@R3# set lo0 unit 3 family inet address 172.16.3.3/32
```

2. Configure the AS number.

```
[edit routing-options]
user@R3# set autonomous-system 65001
```

3. Configure the BGP session with the ISP device.

```
[edit protocols bgp group ext]
user@R3# set type external
user@R3# set peer-as 65000
user@R3# set neighbor 10.0.45.1
```

4. Configure the BGP session with Device R2.

```
[edit protocols bgp group int]
user@R3# set type internal
user@R3# set local-address 172.16.3.3
user@R3# set neighbor 172.16.2.2
```

5. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R3# set interface ge-1/2/0.3
user@R3# set interface lo0.3 passive
```

6. Configure the aggregate route.

```
[edit routing-options]
user@R3# set aggregate route 10.200.0.0/16
```

7. Configure the routing policy to advertise the aggregate route.

   The first term in this policy advertises the aggregate route. The second term prevents more specific routes from being advertised.

```
[edit policy-options policy-statement send-aggregate]
user@R3# set term 1 from protocol aggregate
user@R3# set term 1 then accept
user@R3# set term suppress-specific-routes from route-filter 10.200.0.0/16 longer
user@R3# set term suppress-specific-routes then reject
```

8. Apply the aggregate route policy to the external BGP session with Device ISP.

```
[edit protocols bgp group ext]
user@R3# set export send-aggregate
```

9. Configure the routing policy to advertise the default route from Device ISP.

```
[edit policy-options policy-statement send-default]
user@R3# set from route-filter 0.0.0.0/0 exact
user@R3# set then accept
```

10. Apply the default routing policy to OSPF.

```
[edit protocols ospf]
user@R3# set export send-default
```

11. If you are done configuring the device, commit the configuration.

```
[edit]
user@R3# commit
```

Results Confirm your configuration by issuing the **show interfaces**, **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R3# show interfaces
ge-1/2/0 {
   unit 3 {
      description R3->R2;
      family inet {
         address 10.0.2.1/30;
      }
   }
}
ge-1/2/1 {
   unit 6 {
      description R3->ISP;
      family inet {
         address 10.0.45.2/30;
      }
   }
}
lo0 {
   unit 3 {
      family inet {
         address 172.16.3.3/32;
      }
   }
}
user@R3# show protocols
bgp {
   group ext {
      type external;
      export send-aggregate;
      peer-as 65000;
      neighbor 10.0.45.1;
   }
```

 Copyright © 2012, Juniper Networks, Inc.

```
    group int {
        type internal;
        local-address 172.16.3.3;
        neighbor 172.16.2.2;
    }
}
ospf {
    export send-default;
    area 0.0.0.0 {
        interface ge-1/2/0.3;
        interface lo0.3 {
            passive;
        }
    }
}
user@R3# show policy-options
policy-statement send-aggregate {
    term 1 {
        from protocol aggregate;
        then accept;
    }
    term suppress-specific-routes {
        from {
            route-filter 10.200.0.0/16 longer;
        }
        then reject;
    }
}
policy-statement send-default {
    from {
        route-filter 0.0.0.0/0 exact;
    }
    then accept;
}
user@R3# show routing-options
aggregate {
    route 10.200.0.0/16;
}
autonomous-system 65001;
```

Verification

Confirm that the configuration is working properly.

- Verifying That Device R3 Has the Expected Routes on page 103
- Verifying That Device R3 Advertises the Aggregate Route to Device ISP on page 104

Verifying That Device R3 Has the Expected Routes

Purpose Make sure that Device R3 has the specific static routes.

Action

```
user@R3>show route terse protocol bgp

inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
A Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* 0.0.0.0/0          B 170        100            >10.0.45.1       65000 I
* 10.200.1.0/24      B 170        100            >10.0.2.2        I
* 10.200.2.0/24      B 170        100            >10.0.2.2        I
```

Meaning The output shows that Device R3 has the specific routes to the 10.200.1.0/24 and 10.200.2.0/24 networks.

Verifying That Device R3 Advertises the Aggregate Route to Device ISP

Purpose Make sure that Device R3 does not send the specific static routes and only sends the summarized aggregate route.

Action

```
user@R3>show route advertising-protocol bgp 10.0.45.1
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.200.0.0/16           Self                                    I
```

Meaning The output shows that Device R3 sends only the summarized route to Device ISP.

Related Documentation

- Understanding Aggregate Routes on page 91

 Copyright © 2012, Juniper Networks, Inc.

CHAPTER 6

Packet Forwarding Behavior

- Indirect Next Hops and the Packet Forwarding Engine on page 105
- Unicast Reverse-Path-Forwarding Check on page 115

Indirect Next Hops and the Packet Forwarding Engine

- Understanding Indirect Next Hops on page 105
- Example: Optimizing Route Reconvergence by Enabling Indirect Next Hops on the Packet Forwarding Engine on page 106

Understanding Indirect Next Hops

Junos OS supports the concept of an indirect next hop for all routing protocols that support indirectly connected next hops, also known as third-party next hops.

Because routing protocols such as internal BGP (IBGP) can send routing information about indirectly connected routes, Junos OS relies on routes from intra-AS routing protocols (OSPF, IS-IS, RIP, and static) to resolve the best directly connected next hop. The Routing Engine performs route resolution to determine the best directly connected next hop and installs the route to the Packet Forwarding Engine.

By default, Junos OS does not maintain the route for indirect next hop to forwarding next-hop binding on the Packet Forwarding Engine forwarding table. As a result, when a rerouting event occurs, potentially thousands of route to forwarding next-hop bindings must be updated, which increases the route convergence time. Figure 15 on page 105 illustrates the route to forwarding next-hop bindings with indirect next hop disabled.

Figure 15: Route to Forwarding Next-Hop Bindings

You can enable Junos OS to maintain the indirect next hop to forwarding next-hop binding on the Packet Forwarding Engine forwarding table. As a result, fewer route to forwarding next-hop bindings need to be updated, which improves the route convergence time. Figure 16 on page 106 illustrates the route to forwarding next-hop bindings with indirect next hop enabled.

Figure 16: Route to Forwarding Indirect Next-Hop Bindings

Related Documentation

- Example: Optimizing Route Reconvergence by Enabling Indirect Next Hops on the Packet Forwarding Engine on page 106

Example: Optimizing Route Reconvergence by Enabling Indirect Next Hops on the Packet Forwarding Engine

This example shows how to use indirect next hops to promote faster network convergence (for example, in BGP networks) by decreasing the number of forwarding table changes required when a change in the network topology occurs.

- Requirements on page 106
- Overview on page 106
- Configuration on page 107
- Verification on page 114

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, several devices are connected over unequal-cost paths. From Device R1 to Device R2, the path through Device R3 has a higher IGP metric than the path through Device R4. Device R1 has an internal BGP connection to Device R2. Device R0 injects multiple routes into the network, and Device R1 advertises those routes to Device R2. Because Device R2 is not directly connected to Device R1, Device R2's forwarding table contains indirect next hops. An interior gateway protocol, in this case OSPF, is running on the internal links among Devices R1, R2, R3, and R4. Each router is advertising its loopback interface IPv4 address.

On Device R2, the **indirect-next-hop** statement enables Junos OS to maintain the indirect next hop to forwarding next-hop binding on the Packet Forwarding Engine forwarding table. As a result, fewer route to forwarding next-hop bindings need to be updated, which improves the route convergence time if a path fails.

Figure 17 on page 107 shows the sample network.

Copyright © 2012, Juniper Networks, Inc.

The "CLI Quick Configuration" on page 107 section shows the full configuration on all of the devices in Figure 17 on page 107. Otherwise, the example focuses on Device R0, Device R1, and Device R2.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R0

```
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 1.1.0.1/32
set interfaces lo0 unit 1 family inet address 1.1.0.2/32
set interfaces lo0 unit 1 family inet address 1.1.0.3/32
set interfaces lo0 unit 1 family inet address 1.1.0.4/32
set interfaces lo0 unit 1 family inet address 1.1.0.5/32
set interfaces lo0 unit 1 family inet address 1.1.0.6/32
set interfaces lo0 unit 1 family inet address 1.1.0.7/32
set interfaces lo0 unit 1 family inet address 1.1.0.8/32
set interfaces lo0 unit 1 family inet address 1.1.0.9/32
set routing-options static route 0.0.0.0/0 next-hop 10.0.0.2
```

Device R1

```
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces fe-1/2/2 unit 9 family inet address 10.0.0.9/30
set interfaces lo0 unit 2 family inet address 1.1.1.1/32
set protocols bgp export send-local
set protocols bgp export send-static
set protocols bgp group int type internal
set protocols bgp group int local-address 1.1.1.1
set protocols bgp group int neighbor 2.2.2.2
set protocols ospf area 0.0.0.0 interface fe-1/2/1.5
set protocols ospf area 0.0.0.0 interface fe-1/2/2.9
set protocols ospf area 0.0.0.0 interface lo0.2
set policy-options policy-statement send-local from protocol local
set policy-options policy-statement send-local from protocol direct
set policy-options policy-statement send-local then accept
set policy-options policy-statement send-static from protocol static
set policy-options policy-statement send-static then accept
set routing-options static route 1.1.0.2/32 next-hop 10.0.0.1
```

Copyright © 2012, Juniper Networks, Inc.

```
set routing-options static route 1.1.0.1/32 next-hop 10.0.0.1
set routing-options static route 1.1.0.3/32 next-hop 10.0.0.1
set routing-options static route 1.1.0.4/32 next-hop 10.0.0.1
set routing-options static route 1.1.0.5/32 next-hop 10.0.0.1
set routing-options static route 1.1.0.6/32 next-hop 10.0.0.1
set routing-options static route 1.1.0.7/32 next-hop 10.0.0.1
set routing-options static route 1.1.0.8/32 next-hop 10.0.0.1
set routing-options static route 1.1.0.9/32 next-hop 10.0.0.1
set routing-options autonomous-system 65500
```

Device R2

```
set interfaces fe-1/2/0 unit 14 family inet address 10.0.0.14/30
set interfaces fe-1/2/1 unit 18 family inet address 10.0.0.18/30
set interfaces fe-1/2/2 unit 21 family inet
set interfaces lo0 unit 3 family inet address 2.2.2.2/32
set protocols bgp export send-local
set protocols bgp group int type internal
set protocols bgp group int local-address 2.2.2.2
set protocols bgp group int family inet unicast
set protocols bgp group int family inet-vpn unicast
set protocols bgp group int neighbor 1.1.1.1
set protocols ospf area 0.0.0.0 interface fe-1/2/0.14
set protocols ospf area 0.0.0.0 interface fe-1/2/1.18
set protocols ospf area 0.0.0.0 interface lo0.3
set policy-options policy-statement send-local from protocol local
set policy-options policy-statement send-local from protocol direct
set policy-options policy-statement send-local then accept
set routing-options autonomous-system 65500
set routing-options forwarding-table indirect-next-hop
```

Device R3

```
set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces fe-1/2/1 unit 13 family inet address 10.0.0.13/30
set interfaces lo0 unit 4 family inet address 3.3.3.3/32
set protocols ospf area 0.0.0.0 interface fe-1/2/0.6 metric 5000
set protocols ospf area 0.0.0.0 interface fe-1/2/1.13 metric 5000
set protocols ospf area 0.0.0.0 interface lo0.4
```

Device R4

```
set interfaces fe-1/2/0 unit 10 family inet address 10.0.0.10/30
set interfaces fe-1/2/1 unit 17 family inet address 10.0.0.17/30
set interfaces lo0 unit 5 family inet address 4.4.4.4/32
set protocols ospf area 0.0.0.0 interface fe-1/2/0.10
set protocols ospf area 0.0.0.0 interface fe-1/2/1.17
set protocols ospf area 0.0.0.0 interface lo0.5
```

Device R5

```
set interfaces fe-1/2/0 unit 22 family inet address 10.0.0.22/30
set interfaces lo0 unit 6 family inet address 5.5.5.5/32
```


Copyright © 2012, Juniper Networks, Inc.

Configuring Device R0

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device R0:

1. Configure the interfaces, including multiple routes that can be injected into the network for demonstration purposes.

```
[edit interfaces]
user@R0# set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30

user@R0# set interfaces lo0 unit 1 family inet address 1.1.0.1/32
user@R0# set interfaces lo0 unit 1 family inet address 1.1.0.2/32
user@R0# set interfaces lo0 unit 1 family inet address 1.1.0.3/32
user@R0# set interfaces lo0 unit 1 family inet address 1.1.0.4/32
user@R0# set interfaces lo0 unit 1 family inet address 1.1.0.5/32
user@R0# set interfaces lo0 unit 1 family inet address 1.1.0.6/32
user@R0# set interfaces lo0 unit 1 family inet address 1.1.0.7/32
user@R0# set interfaces lo0 unit 1 family inet address 1.1.0.8/32
user@R0# set interfaces lo0 unit 1 family inet address 1.1.0.9/32
```

2. Configure a static default route for network reachability.

```
[edit routing-options]
user@R0# set routing-options static route 0.0.0.0/0 next-hop 10.0.0.2
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@R0# commit
```

Configuring Device R1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device R1:

1. Configure the interfaces, including multiple routes that can be injected into the network for demonstration purposes.

```
[edit interfaces]
user@R1# set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
user@R1# set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
user@R1# set interfaces fe-1/2/2 unit 9 family inet address 10.0.0.9/30

user@R1# set interfaces lo0 unit 2 family inet address 1.1.1.1/32
```

2. Configure BGP.

```
[edit routing-options]
user@R1# set protocols bgp export send-local
```

```
user@R1# set protocols bgp export send-static
user@R1# set protocols bgp group int type internal
user@R1# set protocols bgp group int local-address 1.1.1.1
user@R1# set protocols bgp group int neighbor 2.2.2.2
```

3. Configure OSPF.

```
[edit protocols]
user@R1# set protocols ospf area 0.0.0.0 interface fe-1/2/1.5
user@R1# set protocols ospf area 0.0.0.0 interface fe-1/2/2.9
user@R1# set protocols ospf area 0.0.0.0 interface lo0.2
```

4. Configure the routing policies.

```
user@R1# set policy-options policy-statement send-local from protocol local
user@R1# set policy-options policy-statement send-local from protocol direct
user@R1# set policy-options policy-statement send-local then accept

user@R1# set policy-options policy-statement send-static from protocol static
user@R1# set policy-options policy-statement send-static then accept
```

5. Configure a set of static routes to the set of interfaces configured on Device R0.

```
user@R1# set routing-options static route 1.1.0.2/32 next-hop 10.0.0.1
user@R1# set routing-options static route 1.1.0.1/32 next-hop 10.0.0.1
user@R1# set routing-options static route 1.1.0.3/32 next-hop 10.0.0.1
user@R1# set routing-options static route 1.1.0.4/32 next-hop 10.0.0.1
user@R1# set routing-options static route 1.1.0.5/32 next-hop 10.0.0.1
user@R1# set routing-options static route 1.1.0.6/32 next-hop 10.0.0.1
user@R1# set routing-options static route 1.1.0.7/32 next-hop 10.0.0.1
user@R1# set routing-options static route 1.1.0.8/32 next-hop 10.0.0.1
user@R1# set routing-options static route 1.1.0.9/32 next-hop 10.0.0.1
```

6. Configure the autonomous system (AS) identifier.

```
user@R1# set routing-options autonomous-system 65500
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@R1# commit
```

Configuring Device R2

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device R2:

1. Configure the interfaces, including multiple routes that can be injected into the network for demonstration purposes.

```
[edit interfaces]
user@R2# set interfaces fe-1/2/0 unit 14 family inet address 10.0.0.14/30
user@R2# set interfaces fe-1/2/1 unit 18 family inet address 10.0.0.18/30
user@R2# set interfaces fe-1/2/2 unit 21 family inet address 10.0.0.21/30;
```


Copyright © 2012, Juniper Networks, Inc.

```
user@R2# set interfaces lo0 unit 3 family inet address 2.2.2.2/32
```

2. Configure BGP.

```
[edit routing-options]
user@R2# set protocols bgp export send-local
user@R2# set protocols bgp group int type internal
user@R2# set protocols bgp group int local-address 2.2.2.2
user@R2# set protocols bgp group int family inet unicast
user@R2# set protocols bgp group int family inet-vpn unicast
user@R2# set protocols bgp group int neighbor 1.1.1.1
```

3. Configure OSPF.

```
[edit protocols]
user@R2# set protocols ospf area 0.0.0.0 interface fe-1/2/0.14
user@R2# set protocols ospf area 0.0.0.0 interface fe-1/2/1.18
user@R2# set protocols ospf area 0.0.0.0 interface lo0.3
```

4. Configure the routing policies.

```
user@R2# set policy-options policy-statement send-local from protocol local
user@R2# set policy-options policy-statement send-local from protocol direct
user@R2# set policy-options policy-statement send-local then accept
```

5. Configure the AS identifier.

```
user@R2# set routing-options autonomous-system 65500
```

6. Enable indirect next hops in the forwarding plane.

```
user@R2# set routing-options forwarding-table indirect-next-hop
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@R2# commit
```

Results Confirm your configuration by issuing the **show interfaces**, **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device R0

```
user@R0# show interfaces
fe-1/2/0 {
   unit 1 {
      family inet {
         address 10.0.0.1/30;
      }
   }
}
lo0 {
   unit 1 {
      family inet {
         address 1.1.0.1/32;
         address 1.1.0.2/32;
         address 1.1.0.3/32;
         address 1.1.0.4/32;
         address 1.1.0.5/32;
         address 1.1.0.6/32;
```

```
            address 1.1.0.7/32;
            address 1.1.0.8/32;
            address 1.1.0.9/32;
        }
    }
}

user@R0# show routing-options
static {
    route 0.0.0.0/0 next-hop 10.0.0.2;
}
```

Device R1

```
user@R1# show interfaces
fe-1/2/0 {
    unit 2 {
        family inet {
            address 10.0.0.2/30;
        }
    }
}
fe-1/2/1 {
    unit 5 {
        family inet {
            address 10.0.0.5/30;
        }
    }
}
fe-1/2/2 {
    unit 9 {
        family inet {
            address 10.0.0.9/30;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 1.1.1.1/32;
        }
    }
}

user@R1# show protocols
bgp {
    export [ send-local send-static ];
    group int {
        type internal;
        local-address 1.1.1.1;
        neighbor 2.2.2.2;
    }
}
ospf {
    area 0.0.0.0 {
        interface fe-1/2/1.5;
        interface fe-1/2/2.9;
        interface lo0.2;
```


Copyright © 2012, Juniper Networks, Inc.

```
    }
}

user@R1# show policy-options
policy-statement send-local {
    from protocol [ local direct ];
    then accept;
}
policy-statement send-static {
    from protocol static;
    then accept;
}

user@R1# show routing-options
static {
    route 1.1.0.2/32 next-hop 10.0.0.1;
    route 1.1.0.1/32 next-hop 10.0.0.1;
    route 1.1.0.3/32 next-hop 10.0.0.1;
    route 1.1.0.4/32 next-hop 10.0.0.1;
    route 1.1.0.5/32 next-hop 10.0.0.1;
    route 1.1.0.6/32 next-hop 10.0.0.1;
    route 1.1.0.7/32 next-hop 10.0.0.1;
    route 1.1.0.8/32 next-hop 10.0.0.1;
    route 1.1.0.9/32 next-hop 10.0.0.1;
}
autonomous-system 65500;
```

Device R2

```
user@R2# show interfaces
fe-1/2/0 {
    unit 14 {
        family inet {
            address 10.0.0.14/30;
        }
    }
}
fe-1/2/1 {
    unit 18 {
        family inet {
            address 10.0.0.18/30;
        }
    }
}
fe-1/2/2 {
    unit 21 {
        family inet {
            address 10.0.0.21/30
        }
    }
}
lo0 {
    unit 3 {
        family inet {
            address 2.2.2.2/32;
        }
    }
}
```

```
user@R2# show protocols
bgp {
  export send-local;
  group int {
    type internal;
    local-address 2.2.2.2;
    family inet {
      unicast;
    }
    family inet-vpn {
      unicast;
    }
    neighbor 1.1.1.1;
  }
}
ospf {
  area 0.0.0.0 {
    interface fe-1/2/0.14;
    interface fe-1/2/1.18;
    interface lo0.3;
  }
}
```

```
user@R2# show policy-options
policy-statement send-local {
  from protocol [ local direct ];
  then accept;
}
```

```
user@R2# show routing-options
autonomous-system 65500;
forwarding-table {
  indirect-next-hop;
}
```

Configure Device R3, Device R4, and Device R5, as shown in "CLI Quick Configuration" on page 107.

Verification

Confirm that the configuration is working properly.

Verifying That the Routes Have the Expected Indirect-Next-Hop Flag

Purpose Make sure that Device R2 is configured to maintain the indirect next hop to forwarding next-hop binding on the Packet Forwarding Engine forwarding table.

Action

```
user@R2> show krt indirect-next-hop
Indirect Nexthop:
Index: 262143 Protocol next-hop address: 1.1.1.1
  RIB Table: inet.0
  Policy Version: 0                    References: 4
  Locks: 2                             0x9290488
  Flags: 0x1
  Ref RIB Table: unknown
        Next hop: 10.0.0.17 via fe-1/2/1.18
Indirect Nexthop:
```

 Copyright © 2012, Juniper Networks, Inc.

```
Index: 262142 Protocol next-hop address: 10.0.0.1
  RIB Table: inet.0
  Policy Version: 0                    References: 9
  Locks: 2                             0x9290570
  Flags: 0x1
  Ref RIB Table: unknown
        Next hop: 10.0.0.17 via fe-1/2/1.18
```

Meaning The **0x1** flag in the output indicates that Device R2 is configured to maintain the indirect next hop to forwarding next-hop binding on the Packet Forwarding Engine forwarding table. When the **indirect-next-hop** statement is deleted or deactivated, this flag changes to **0x0**.

NOTE: The **show krt indirect-next-hop** command is hidden and is therefore undocumented. The **show krt indirect-next-hop** command is shown here because this is the only command that verifies the indirect next-hop feature. The best verification method is, of course, monitoring network performance during reconvergence after a path failure.

Related Documentation

- Understanding Indirect Next Hops on page 105

Unicast Reverse-Path-Forwarding Check

- Understanding Unicast Reverse Path Forwarding on page 115
- Example: Configuring Unicast Reverse-Path-Forwarding Check on page 116

Understanding Unicast Reverse Path Forwarding

IP spoofing can occur during a denial-of-service (DoS) attack. IP spoofing allows an intruder to pass IP packets to a destination as genuine traffic, when in fact the packets are not actually meant for the destination. This type of spoofing is harmful because it consumes the destination's resources.

A unicast reverse-path-forwarding (RPF) check is a tool to reduce forwarding of IP packets that might be spoofing an address. A unicast RPF check performs a route table lookup on an IP packet's source address, and checks the incoming interface. The router determines whether the packet is arriving from a path that the sender would use to reach the destination. If the packet is from a valid path, the router forwards the packet to the destination address. If it is not from a valid path, the router discards the packet. Unicast RPF is supported for the IPv4 and IPv6 protocol families, as well as for the virtual private network (VPN) address family.

NOTE: Reverse path forwarding is not supported on the interfaces you configure as tunnel sources. This affects only the transit packets exiting the tunnel.

Related Documentation

- Example: Configuring Unicast Reverse-Path-Forwarding Check on page 116

Example: Configuring Unicast Reverse-Path-Forwarding Check

Unicast reverse path forwarding (RPF) helps protect against DoS and DDoS attacks by verifying the unicast source address of each packet that arrives on an ingress interface where unicast RPF is enabled.

This example shows how to help defend ingress interfaces against denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks by configuring unicast RPF to filter incoming traffic.

- Requirements on page 116
- Overview on page 116
- Configuration on page 117
- Verification on page 122

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

Large amounts of unauthorized traffic such as attempts to flood a network with fake (bogus) service requests in a DoS attack can consume network resources and deny service to legitimate users. One way to help prevent DoS and DDoS attacks is to verify that incoming traffic originates from legitimate network sources.

Unicast RPF helps ensure that a traffic source is legitimate (authorized) by comparing the source address of each packet that arrives on an interface to the forwarding table entry for its source address. If the device uses the same interface that the packet arrived on to reply to the packet's source, this verifies that the packet originated from an authorized source, and the device forwards the packet. If the device does not use the same interface that the packet arrived on to reply to the packet's source, the packet might have originated from an unauthorized source, and the device discards the packet.

In this example, Device B has unicast RPF configured. Device A is using OSPF to advertise a prefix for the link that connects to Device D. OSPF is enabled on the links between Device B and Device C and the links between Device A and Device C, but not on the links between Device A and Device B. Therefore, Device B learns about the route to Device D through Device C.

This example also includes a fail filter. When a packet fails the unicast RPF check, the fail filter is evaluated to determine if the packet should be accepted anyway. The fail filter in this example allows Device B's interfaces to accept Dynamic Host Configuration Protocol (DHCP) packets. The filter accepts all packets with a source address of 0.0.0.0 and a destination address of 255.255.255.255.

Figure 18 on page 117 shows the sample network.

Copyright © 2012, Juniper Networks, Inc.

Figure 18: Unicast RPF Sample Topoolgy

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device A

```
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces fe-0/0/2 unit 5 family inet address 10.0.0.5/30
set interfaces fe-0/0/1 unit 17 family inet address 10.0.0.17/30
set interfaces fe-0/1/1 unit 25 family inet address 10.0.0.25/30
set interfaces fe-1/1/1 unit 29 family inet address 10.0.0.29/30
set protocols ospf export send-direct
set protocols ospf area 0.0.0.0 interface fe-0/1/1.25
set protocols ospf area 0.0.0.0 interface fe-1/1/1.29
set policy-options policy-statement send-direct from protocol direct
set policy-options policy-statement send-direct from route-filter 10.0.0.16/30 exact
set policy-options policy-statement send-direct then accept
```

Device B

```
set interfaces fe-1/2/0 unit 2 family inet rpf-check fail-filter rpf-special-case-dhcp
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/1/1 unit 6 family inet rpf-check fail-filter rpf-special-case-dhcp
set interfaces fe-1/1/1 unit 6 family inet address 10.0.0.6/30
set interfaces fe-0/1/1 unit 9 family inet rpf-check fail-filter rpf-special-case-dhcp
set interfaces fe-0/1/1 unit 9 family inet address 10.0.0.9/30
set interfaces fe-0/1/0 unit 13 family inet rpf-check fail-filter rpf-special-case-dhcp
set interfaces fe-0/1/0 unit 13 family inet address 10.0.0.13/30
set protocols ospf area 0.0.0.0 interface fe-0/1/1.9
set protocols ospf area 0.0.0.0 interface fe-0/1/0.13
set routing-options forwarding-table unicast-reverse-path active-paths
set firewall filter rpf-special-case-dhcp term allow-dhcp from source-address 0.0.0.0/32
set firewall filter rpf-special-case-dhcp term allow-dhcp then count rpf-dhcp-traffic
set firewall filter rpf-special-case-dhcp term allow-dhcp then accept
set firewall filter rpf-special-case-dhcp term default then log
set firewall filter rpf-special-case-dhcp term default then reject
```

Device C

```
set interfaces fe-1/2/0 unit 10 family inet address 10.0.0.10/30
set interfaces fe-0/0/2 unit 14 family inet address 10.0.0.14/30
set interfaces fe-1/0/2 unit 21 family inet address 10.0.0.21/30
set interfaces fe-1/2/2 unit 26 family inet address 10.0.0.26/30
set interfaces fe-1/2/1 unit 30 family inet address 10.0.0.30/30
```

Copyright © 2012, Juniper Networks, Inc.

```
set protocols ospf area 0.0.0.0 interface fe-1/2/0.10
set protocols ospf area 0.0.0.0 interface fe-0/0/2.14
set protocols ospf area 0.0.0.0 interface fe-1/2/2.26
set protocols ospf area 0.0.0.0 interface fe-1/2/1.30
```

Device D

```
set interfaces fe-1/2/0 unit 18 family inet address 10.0.0.18/30
```

Device E

```
set interfaces fe-1/2/0 unit 22 family inet address 10.0.0.22/30
```

Configuring Device A

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device A:

1. Configure the interfaces.

```
[edit interfaces]
user@A# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

user@A# set fe-0/0/2 unit 5 family inet address 10.0.0.5/30

user@A# set fe-0/0/1 unit 17 family inet address 10.0.0.17/30

user@A# set fe-0/1/1 unit 25 family inet address 10.0.0.25/30

user@A# set fe-1/1/1 unit 29 family inet address 10.0.0.29/30
```

2. Configure OSPF.

```
[edit protocols ospf]
user@A# set export send-direct
user@A# set area 0.0.0.0 interface fe-0/1/1.25
user@A# set area 0.0.0.0 interface fe-1/1/1.29
```

3. Configure the routing policy.

```
[edit policy-options policy-statement send-direct]
user@A# set from protocol direct
user@A# set from route-filter 10.0.0.16/30 exact
user@A# set then accept
```

4. If you are done configuring Device A, commit the configuration.

```
[edit]
user@A# commit
```


Copyright © 2012, Juniper Networks, Inc.

Configuring Device B

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device B:

1. Configure the interfaces.

```
[edit interfaces]
user@B# set fe-1/2/0 unit 2 family inet address 10.0.0.2/30

user@B# set fe-1/1/1 unit 6 family inet address 10.0.0.6/30

user@B# set fe-0/1/1 unit 9 family inet address 10.0.0.9/30

user@B# set fe-0/1/0 unit 13 family inet address 10.0.0.13/30
```

2. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@B# set interface fe-0/1/1.9
user@B# set interface fe-0/1/0.13
```

3. Configure unicast RPF, and apply the optional fail filter.

```
[edit interfaces]
user@B# set fe-1/2/0 unit 2 family inet rpf-check fail-filter rpf-special-case-dhcp

user@B# set fe-1/1/1 unit 6 family inet rpf-check fail-filter rpf-special-case-dhcp

user@B# set fe-0/1/1 unit 9 family inet rpf-check fail-filter rpf-special-case-dhcp

user@B# set fe-0/1/0 unit 13 family inet rpf-check fail-filter rpf-special-case-dhcp
```

4. (Optional) Configure the fail filter that gets evaluated if a packet fails the RPF check.

```
[edit firewall filter rpf-special-case-dhcp]
user@B# set term allow-dhcp from source-address 0.0.0.0/32
user@B# set term allow-dhcp then count rpf-dhcp-traffic
user@B# set term allow-dhcp then accept
user@B# set term default then log
user@B# set term default then reject
```

5. (Optional) Configure only active paths to be considered in the RPF check.

   This is the default behavior.

```
[edit routing-options forwarding-table]
user@B# set unicast-reverse-path active-paths
```

6. If you are done configuring Device B, commit the configuration.

```
[edit]
user@B# commit
```

Results Confirm your configuration by issuing the **show firewall**, **show interfaces**, **show protocols**, **show routing-options**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device A

```
user@A# show interfaces
fe-1/2/0 {
   unit 1 {
      family inet {
         address 10.0.0.1/30;
      }
   }
}
fe-0/0/2 {
   unit 5 {
      family inet {
         address 10.0.0.5/30;
      }
   }
}
fe-0/0/1 {
   unit 17 {
      family inet {
         address 10.0.0.17/30;
      }
   }
}
fe-0/1/1 {
   unit 25 {
      family inet {
         address 10.0.0.25/30;
      }
   }
}
fe-1/1/1 {
   unit 29 {
      family inet {
         address 10.0.0.29/30;
      }
   }
}

user@A# show protocols
ospf {
   export send-direct;
   area 0.0.0.0 {
      interface fe-0/1/1.25;
      interface fe-1/1/1.29;
   }
}

user@A# show policy-options
policy-statement send-direct {
   from {
      protocol direct;
      route-filter 10.0.0.16/30 exact;
```

 Copyright © 2012, Juniper Networks, Inc.

```
    }
    then accept;
}
```

Device B

```
user@B# show firewall
filter rpf-special-case-dhcp {
    term allow-dhcp {
        from {
            source-address {
                0.0.0.0/32;
            }
        }
        then {
            count rpf-dhcp-traffic;
            accept;
        }
    }
    term default {
        then {
            log;
            reject;
        }
    }
}
user@B# show interfaces
fe-1/2/0 {
    unit 2 {
        family inet {
            rpf-check fail-filter rpf-special-case-dhcp;
            address 10.0.0.2/30;
        }
    }
}
fe-1/1/1 {
    unit 6 {
        family inet {
            rpf-check fail-filter rpf-special-case-dhcp;
            address 10.0.0.6/30;
        }
    }
}
fe-0/1/1 {
    unit 9 {
        family inet {
            rpf-check fail-filter rpf-special-case-dhcp;
            address 10.0.0.9/30;
        }
    }
}
fe-0/1/0 {
    unit 13 {
        family inet {
            rpf-check fail-filter rpf-special-case-dhcp;
            address 10.0.0.13/30;
        }
    }
```

```
}

user@B# show protocols
ospf {
  area 0.0.0.0 {
    interface fe-0/1/1.9;
    interface fe-0/1/0.13;
  }
}

user@B# show routing-options
forwarding-table {
  unicast-reverse-path active-paths;
}
```

Enter the configurations on Device C, Device D, and Device E, as shown in "CLI Quick Configuration" on page 117.

Verification

Confirm that the configuration is working properly.

- Confirm That Unicast RPF Is Enabled on page 122
- Confirm That the Source Addresses Are Blocked on page 123
- Confirm That the Source Addresses Are Unblocked on page 123

Confirm That Unicast RPF Is Enabled

Purpose Make sure that the interfaces on Device B have unicast RPF enabled.

Action

```
user@B> show interfaces fe-0/1/0.13 extensive
 Logical interface fe-0/1/0.13 (Index 73) (SNMP ifIndex 553) (Generation 208)
    Flags: SNMP-Traps 0x4000 Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :               999390
     Output bytes  :              1230122
     Input  packets:                12563
     Output packets:                12613
    Local statistics:
     Input  bytes  :               998994
     Output bytes  :              1230122
     Input  packets:                12563
     Output packets:                12613
    Transit statistics:
     Input  bytes  :                  396                    0 bps
     Output bytes  :                    0                    0 bps
     Input  packets:                    0                    0 pps
     Output packets:                    0                    0 pps
    Protocol inet, MTU: 1500, Generation: 289, Route table: 22
      Flags: Sendbcast-pkt-to-re, uRPF
      RPF Failures: Packets: 0, Bytes: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.0.0.12/30, Local: 10.0.0.13, Broadcast: 10.0.0.15,
Generation: 241
```

Meaning The **uRPF** flag confirms that unicast RPF is enabled on this interface.

 Copyright © 2012, Juniper Networks, Inc.

Confirm That the Source Addresses Are Blocked

Purpose Use the **ping** command to make sure that Device B blocks traffic from unexpected source addresses.

Action From Device A, ping Device B's interfaces, using 10.0.0.17 as the source address.

```
user@A> ping 10.0.0.6 source 10.0.0.17
PING 10.0.0.6 (10.0.0.6): 56 data bytes
^C
--- 10.0.0.6 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
```

Meaning As expected, the ping operation fails.

Confirm That the Source Addresses Are Unblocked

Purpose Use the **ping** command to make sure that Device B does not block traffic when the RPF check is deactivated.

Action

1. Deactivate the RPF check on one of the interfaces.
2. Rerun the ping operation.

```
user@B> deactivate interfaces fe-1/1/1.6 family inet rpf-check
```

```
user@A> ping 10.0.0.6 source 10.0.0.17
PING 10.0.0.2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=63 time=1.316 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=63 time=1.263 ms
^C
--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.263/1.289/1.316/0.027 ms
```

Meaning As expected, the ping operation succeeds.

Related Documentation

- Understanding Unicast Reverse Path Forwarding on page 115

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 7

Martian Addresses

- Understanding Martian Addresses on page 125
- Example: Configuring Martian Addresses on page 126

Understanding Martian Addresses

Martian addresses are host or network addresses about which all routing information is ignored. When received by the routing device, these routes are ignored. They commonly are sent by improperly configured systems on the network and have destination addresses that are obviously invalid.

In IPv6, the loopback address and the multicast resolve and discard routes are the default martian addresses.

In Junos OS Release 10.4R5 and later, the reserved IPv6 multicast address space (ff00::/8 and ff02::/16) is added to the list of martian addresses.

In Junos OS Release 9.6 and later, you can configure Class E addresses on interfaces. Class E addresses are treated like any other unicast address for the purpose of forwarding. To allow Class E addresses to be configured on interfaces, you must remove the Class E prefix from the list of martian addresses. To remove the Class E prefix from the list of martian addresses include the **martians 240/4 orlonger allow** statement at the **[edit routing-options]** hierarchy level.

To view the default and configured martian routes, run the **show route martians** command.

IPv4 Martian Addresses

```
user@host> show route martians table inet.

inet.0:
             0.0.0.0/0 exact -- allowed
             0.0.0.0/8 orlonger -- disallowed
             127.0.0.0/8 orlonger -- disallowed
             192.0.0.0/24 orlonger -- disallowed
             240.0.0.0/4 orlonger -- disallowed
             224.0.0.0/4 exact -- disallowed
             224.0.0.0/24 exact -- disallowed

inet.1:
             0.0.0.0/0 exact -- allowed
             0.0.0.0/8 orlonger -- disallowed
             127.0.0.0/8 orlonger -- disallowed
             192.0.0.0/24 orlonger -- disallowed
```

Copyright © 2012, Juniper Networks, Inc.

```
                240.0.0.0/4 orlonger -- disallowed

inet.2:
                0.0.0.0/0 exact -- allowed
                0.0.0.0/8 orlonger -- disallowed
                127.0.0.0/8 orlonger -- disallowed
                192.0.0.0/24 orlonger -- disallowed
                240.0.0.0/4 orlonger -- disallowed
                224.0.0.0/4 exact -- disallowed
                224.0.0.0/24 exact -- disallowed

inet.3:
                0.0.0.0/0 exact -- allowed
                0.0.0.0/8 orlonger -- disallowed
                127.0.0.0/8 orlonger -- disallowed
                192.0.0.0/24 orlonger -- disallowed
                240.0.0.0/4 orlonger -- disallowed
                224.0.0.0/4 exact -- disallowed
                224.0.0.0/24 exact -- disallowed
```

IPv6 Martian Addresses

```
user@host> show route martians table inet6
inet6.0:
                ::1/128 exact -- disallowed
                ff00::/8 exact -- disallowed
                ff02::/16 exact -- disallowed

inet6.1:
                ::1/128 exact -- disallowed

inet6.2:
                ::1/128 exact -- disallowed
                ff00::/8 exact -- disallowed
                ff02::/16 exact -- disallowed

inet6.3:
                ::1/128 exact -- disallowed
                ff00::/8 exact -- disallowed
                ff02::/16 exact -- disallowed
```

Related Documentation

- Example: Configuring Martian Addresses on page 126

Example: Configuring Martian Addresses

This example shows how to remove the Class E prefix from the list of martian addresses.

- Requirements on page 126
- Overview on page 127
- Configuration on page 127
- Verification on page 128

Requirements

No special configuration beyond device initialization is required before configuring this example.

Copyright © 2012, Juniper Networks, Inc.

Overview

In this example, Junos OS defaults are modified to allow the 240.0.0.0/4 address block. This block of addresses is known as the experimental Class E addresses. In Junos OS Release 9.6 and later, you can configure Class E addresses on interfaces and use them for forwarding traffic. However, to do this, you must first allow routing on this address block.

This example also shows how to modify the martian addresses in the IPv6 routing table, **inet6.0**.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set routing-options rib inet.1 martians 240.0.0.0/4 orlonger allow
set routing-options rib inet6.0 martians fd00::/8 orlonger
set routing-options rib inet.3 martians 240.0.0.0/4 orlonger allow
set routing-options rib inet.2 martians 240.0.0.0/4 orlonger allow
set routing-options martians 240.0.0.0/4 orlonger allow
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure martian routes:

1. Allow Class C addresses in the default unicast routing table.

   ```
   [edit routing-options]
   user@host# set martians 240.0.0.0/4 orlonger allow
   ```

2. Allow Class C addresses in the routing table that is used for the IPv4 multicast forwarding cache.

   ```
   [edit routing-options]
   user@host# set rib inet.1 martians 240.0.0.0/4 orlonger allow
   ```

3. Allow Class C addresses in the routing table that is used for multicast reverse path forwarding (RPF) lookup.

   ```
   [edit routing-options]
   user@host# set rib inet.2 martians 240.0.0.0/4 orlonger allow
   ```

4. Allow Class C addresses in the routing table that stores MPLS LSP information.

   ```
   [edit routing-options]
   user@host# set rib inet.3 martians 240.0.0.0/4 orlonger allow
   ```

5. Add a disallowed martian route to the IPv6 unicast routing table.

   ```
   [edit routing-options]
   user@host# set rib inet6.0 martians fd00::/8 orlonger
   ```

Copyright © 2012, Juniper Networks, Inc.

6. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results Confirm your configuration by issuing the **show routing-options** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show routing-options
rib inet.1 {
   martians {
      240.0.0.0/4 orlonger allow;
   }
}
rib inet6.0 {
   martians {
      fd00::/8 orlonger;
   }
}
rib inet.3 {
   martians {
      240.0.0.0/4 orlonger allow;
   }
}
rib inet.2 {
   martians {
      240.0.0.0/4 orlonger allow;
   }
}
martians {
   240.0.0.0/4 orlonger allow;
}
```

Verification

Confirm that the configuration is working properly.

- Verifying That the 240.0.0.0/4 Routes Are Now Accepted on page 128
- Verifying That the fd00::/8 Routes Are Now Rejected on page 129

Verifying That the 240.0.0.0/4 Routes Are Now Accepted

Purpose Make sure that the 240.0.0.0/4 route appears in the routing tables as allowed.

Action

```
user@host> show route martians table inet.
inet.0:
             0.0.0.0/0 exact -- allowed
             0.0.0.0/8 orlonger -- disallowed
             127.0.0.0/8 orlonger -- disallowed
             192.0.0.0/24 orlonger -- disallowed
             240.0.0.0/4 orlonger -- allowed
             224.0.0.0/4 exact -- disallowed
             224.0.0.0/24 exact -- disallowed
```


Copyright © 2012, Juniper Networks, Inc.

```
inet.1:
            0.0.0.0/0 exact -- allowed
            0.0.0.0/8 orlonger -- disallowed
            127.0.0.0/8 orlonger -- disallowed
            192.0.0.0/24 orlonger -- disallowed
            240.0.0.0/4 orlonger -- allowed

inet.2:
            0.0.0.0/0 exact -- allowed
            0.0.0.0/8 orlonger -- disallowed
            127.0.0.0/8 orlonger -- disallowed
            192.0.0.0/24 orlonger -- disallowed
            240.0.0.0/4 orlonger -- allowed
            224.0.0.0/4 exact -- disallowed
            224.0.0.0/24 exact -- disallowed

inet.3:
            0.0.0.0/0 exact -- allowed
            0.0.0.0/8 orlonger -- disallowed
            127.0.0.0/8 orlonger -- disallowed
            192.0.0.0/24 orlonger -- disallowed
            240.0.0.0/4 orlonger -- allowed
            224.0.0.0/4 exact -- disallowed
            224.0.0.0/24 exact -- disallowed
```

Meaning The output shows that the 240.0.0.0/4 route is allowed.

Verifying That the fd00::/8 Routes Are Now Rejected

Purpose Make sure that the fd00::/8 route appears in the IPv6 unicast routing table as disallowed.

Action

```
user@host> show route martians table inet6.0
inet6.0:
            ::1/128 exact -- disallowed
            ff00::/8 exact -- disallowed
            ff02::/16 exact -- disallowed
            fd00::/8 orlonger -- disallowed
```

Meaning The output shows that the fd00::/8 route is disallowed.

Related Documentation

- Example: Creating an Interface on a Logical System on page 33
- Example: Configuring an OSPF Default Route Policy on Logical Systems

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 8

RIP

- RIP Overview on page 131
- RIPng Overview on page 135
- RIP Configuration Overview on page 137
- Basic RIP Routing on page 137
- RIP Traffic Control Through Metrics on page 144
- RIP Authentication on page 148
- Verifying a RIP Configuration on page 150
- RIP Demand Circuits Overview on page 153
- Example: Configuring RIP Demand Circuits on page 155

RIP Overview

In a RIP network, each router's forwarding table is distributed among the nodes through the flooding of routing table information. Because topology changes are flooded throughout the network, every node maintains the same list of destinations. Packets are then routed to these destinations based on path-cost calculations done at each node in the network.

NOTE: In general, the term *RIP* refers to RIP version 1 and RIP version 2.

This topic contains the following sections:

- Distance-Vector Routing Protocols on page 131
- Maximizing Hop Count on page 132
- RIP Packets on page 133
- Split Horizon and Poison Reverse Efficiency Techniques on page 133
- Limitations of Unidirectional Connectivity on page 134

Distance-Vector Routing Protocols

Distance-vector routing protocols transmit routing information that includes a distance vector, typically expressed as the number of hops to the destination. This information is

flooded out all protocol-enabled interfaces at regular intervals (every 30 seconds in the case of RIP) to create a network map that is stored in each node's local topology database. Figure 19 on page 132 shows how distance-vector routing works.

Figure 19: Distance-Vector Protocol

In Figure 19 on page 132, Routers A and B have RIP enabled on adjacent interfaces. Router A has known RIP neighbors Routers C, D, and E, which are 1, 2, and 3 hops away, respectively. Router B has known RIP neighbors Routers X, Y, and Z, which are 1, 2, and 3 hops away, respectively. Every 30 seconds, each router floods its entire routing table information out all RIP-enabled interfaces. In this case, flooding exchanges routing table information across the RIP link.

When Router A receives routing information from Router B, it adds 1 to the hop count to determine the new hop count. For example, Router X has a hop count of 1, but when Router A imports the route to X, the new hop count is 2. The imported route also includes information about where the route was learned, so that the original route is imported as a route to Router X through Router B with a hop count of 2.

When multiple routes to the same host are received, RIP uses the distance-vector algorithm to determine which path to import into the forwarding table. The route with the smallest hop count is imported. If there are multiple routes with the same hop count, all are imported into the forwarding table, and traffic is sent along the paths in round-robin fashion.

Maximizing Hop Count

The successful routing of traffic across a RIP network requires that every node in the network maintain the same view of the topology. Topology information is broadcast between RIP neighbors every 30 seconds. If Router A is many hops away from a new host, Router B, the route to B might take significant time to propagate through the network and be imported into Router A's routing table. If the two routers are 5 hops away from each other, Router A cannot import the route to Router B until 2.5 minutes after Router B is online. For large numbers of hops, the delay becomes prohibitive. To help prevent this delay from growing arbitrarily large, RIP enforces a maximum hop count of 15 hops. Any prefix that is more than 15 hops away is treated as unreachable and assigned a hop count equal to infinity. This maximum hop count is called the *network diameter*.

Copyright © 2012, Juniper Networks, Inc.

RIP Packets

Routing information is exchanged in a RIP network by RIP request and RIP response packets. A router that has just booted can broadcast a RIP request on all RIP-enabled interfaces. Any routers running RIP on those links receive the request and respond by sending a RIP response packet immediately to the router. The response packet contains the routing table information required to build the local copy of the network topology map.

In the absence of RIP request packets, all RIP routers broadcast a RIP response packet every 30 seconds on all RIP-enabled interfaces. The RIP broadcast is the primary way in which topology information is flooded throughout the network.

Once a router learns about a particular destination through RIP, it starts a timer. Every time it receives a new response packet with information about the destination, the router resets the timer to zero. However, if the router receives no updates about a particular destination for 180 seconds, it removes the destination from its RIP routing table.

In addition to the regular transmission of RIP packets every 30 seconds, if a router detects a new neighbor or detects that an interface is unavailable, it generates a triggered update. The new routing information is immediately broadcast out all RIP-enabled interfaces, and the change is reflected in all subsequent RIP response packets.

Split Horizon and Poison Reverse Efficiency Techniques

Because RIP functions by periodically flooding the entire routing table out to the network, it generates a lot of traffic. The split horizon and poison reverse techniques can help reduce the amount of network traffic originated by RIP hosts and make the transmission of routing information more efficient.

If a router receives a set of route advertisements on a particular interface, RIP determines that those advertisements do not need to be retransmitted out the same interface. This technique, known as *split horizon*, helps limit the amount of RIP routing traffic by eliminating information that other neighbors on that interface have already learned. Figure 20 on page 133 shows an example of the split horizon technique.

Figure 20: Split Horizon Example

In Figure 20 on page 133, Router A advertises routes to Routers C, D, and E to Router B. In this example, Router A can reach Router C in 2 hops. When Router A advertises the route to Router B, B imports it as a route to Router C through Router A in 3 hops. If Router B

Copyright © 2012, Juniper Networks, Inc.

then readvertised this route to Router A, A would import it as a route to Router C through Router B in 4 hops. However, the advertisement from Router B to Router A is unnecessary, because Router A can already reach the route in 2 hops. The split horizon technique helps reduce extra traffic by eliminating this type of route advertisement.

Similarly, the poison reverse technique helps to optimize the transmission of routing information and improve the time to reach network convergence. If Router A learns about unreachable routes through one of its interfaces, it advertises those routes as unreachable (hop count of 16) out the same interface. Figure 21 on page 134 shows an example of the poison reverse technique.

Figure 21: Poison Reverse Example

In Figure 21 on page 134, Router A learns through one of its interfaces that routes to Routers C, D, and E are unreachable. Router A readvertises those routes out the same interface as unreachable. The advertisement informs Router B that Hosts C, D, and E are definitely not reachable through Router A.

Limitations of Unidirectional Connectivity

Because RIP processes routing information based solely on the receipt of routing table updates, it cannot ensure bidirectional connectivity. As Figure 22 on page 134 shows, RIP networks are limited by their unidirectional connectivity.

Figure 22: Limitations of Unidirectional Connectivity

In Figure 22 on page 134, Routers A and D flood their routing table information to Router B. Because the path to Router E has the fewest hops when routed through Router A, that

Copyright © 2012, Juniper Networks, Inc.

route is imported into Router B's forwarding table. However, suppose that Router A can transmit traffic but is not receiving traffic from Router B because of an unavailable link or invalid routing policy. If the only route to Router E is through Router A, any traffic destined for Router A is lost, because bidirectional connectivity was never established.

OSPF establishes bidirectional connectivity with a three-way handshake.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Overview on page 3
- RIP Configuration Overview on page 137
- Understanding Basic RIP Routing on page 137
- Understanding RIP Traffic Control with Metrics on page 144
- Understanding RIP Authentication on page 148
- RIPng Overview on page 135
- OSPF Overview on page 160

RIPng Overview

The Routing Information Protocol next generation (RIPng) is an interior gateway protocol (IGP) that uses a distance-vector algorithm to determine the best route to a destination, using hop count as the metric. RIPng exchanges routing information used to compute routes and is intended for IP version 6 (IPv6)-based networks. RIPng is disabled by default.

On devices in secure context, IPv6 is disabled. You must enable IPv6 to use RIPng. For instructions, see the *Junos OS Interfaces Configuration Guide for Security Devices*.

This topic contains the following sections:

- RIPng Protocol Overview on page 135
- RIPng Standards on page 136
- RIPng Packets on page 136

RIPng Protocol Overview

The RIPng IGP uses the Bellman-Ford distance-vector algorithm to determine the best route to a destination, using hop count as the metric. RIPng allows hosts and routers to exchange information for computing routes through an IP-based network. RIPng is intended to act as an IGP for moderately-sized autonomous systems.

RIPng is a distinct routing protocol from RIPv2. The Junos OS implementation of RIPng is similar to RIPv2, but has the following differences:

- RIPng does not need to implement authentication on packets.
- Junos OS does not support multiple instances of RIPng.
- Junos OS does not support RIPng routing table groups.

Copyright © 2012, Juniper Networks, Inc.

RIPng is a UDP-based protocol and uses UDP port 521.

RIPng has the following architectural limitations:

- The longest network path cannot exceed 15 hops (assuming that each network, or hop, has a cost of 1).
- RIPng is prone to routing loops when the routing tables are reconstructed. Especially when RIPng is implemented in large networks that consist of several hundred routers, RIPng might take an extremely long time to resolve routing loops.
- RIPng uses only a fixed metric to select a route. Other IGPs use additional parameters, such as measured delay, reliability, and load.

RIPng Standards

RIPng is defined in the following documents:

- RFC 2080, *RIPng for IPv6*
- RFC 2081, *RIPng Protocol Applicability Statement*

To access Internet Requests for Comments (RFCs) and drafts, see the Internet Engineering Task Force (IETF) website at http://www.ietf.org.

RIPng Packets

A RIPng packet header contains the following fields:

- Command—Indicates whether the packet is a request or response message. Request messages seek information for the router's routing table. Response messages are sent periodically or when a request message is received. Periodic response messages are called update messages. Update messages contain the command and version fields and a set of destinations and metrics.
- Version number—Specifies the version of RIPng that the originating router is running. This is currently set to Version 1.

The rest of the RIPng packet contains a list of routing table entries consisting of the following fields:

- Destination prefix—128-bit IPv6 address prefix for the destination.
- Prefix length—Number of significant bits in the prefix.
- Metric—Value of the metric advertised for the address.
- Route tag—A route attribute that must be advertised and redistributed with the route. Primarily, the route tag distinguishes external RIPng routes from internal RIPng routes when routes must be redistributed across an exterior gateway protocol (EGP).

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Overview on page 3
- RIP Overview on page 131

 Copyright © 2012, Juniper Networks, Inc.

- Configuring RIPng in the *Junos OS Routing Protocols Configuration Guide*
- Minimum RIPng Configuration in the *Junos OS Routing Protocols Configuration Guide*

RIP Configuration Overview

To achieve basic connectivity between all RIP hosts in a RIP network, you enable RIP on every interface that is expected to transmit and receive RIP traffic, as described in the steps that follow.

To configure a RIP network:

1. Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Define RIP groups, which are logical groupings of interfaces, and add interfaces to the groups. Then, configure a routing policy to export directly connected routes and routes learned through RIP routing exchanges. See "Example: Configuring a Basic RIP Network" on page 138.
3. (Optional) Configure metrics to control traffic through the RIP network. See"Example: Controlling Traffic with an Incoming Metric" on page 145 and "Example: Controlling Traffic with an Outgoing Metric" on page 146.
4. (Optional) Configure authentication to ensure that only trusted routers participate in the autonomous system's routing. See"Enabling Authentication with Plain-Text Passwords (CLI Procedure)" on page 149 and "Enabling Authentication with MD5 Authentication (CLI Procedure)" on page 149.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- RIP Overview on page 131
- Verifying a RIP Configuration on page 150

Basic RIP Routing

- Understanding Basic RIP Routing on page 137
- Example: Configuring a Basic RIP Network on page 138

Understanding Basic RIP Routing

RIP is an interior gateway protocol (IGP) that routes packets within a single autonomous system (AS). By default, RIP does not advertise the subnets that are directly connected through the device's interfaces. For traffic to pass through a RIP network, you must create a routing policy to export these routes. Advertising only the direct routes propagates the routes to the immediately adjacent RIP-enabled router only. To propagate all routes through the entire RIP network, you must configure the routing policy to export the routes learned through RIP.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- RIP Overview on page 131
- Example: Configuring a Basic RIP Network on page 138

Example: Configuring a Basic RIP Network

This example shows how to configure a basic RIP network.

- Requirements on page 138
- Overview on page 138
- Configuration on page 139
- Verification on page 141

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you configure a basic RIP network, you create RIP group called **rip-group**, and add the directly connected interfaces to the RIP group. Then you configure a routing policy to advertise direct routes using policy statement **advertise-routes-through-rip**.

By default, Junos OS does not advertise RIP routes, not even routes that are learned through RIP. To advertise RIP routes, you must configure and apply an export routing policy that advertises RIP-learned and direct routes.

In Junos OS, you do not need to configure the RIP version. RIP Version 2 is used by default.

To use RIP on the device, you must configure RIP on all of the RIP interfaces within the network. Figure 23 on page 138 shows the topology used in this example.

Figure 23: Sample RIP Network Topology

"CLI Quick Configuration" on page 139 shows the configuration for all of the devices in Figure 23 on page 138. "Step-by-Step Procedure" on page 139 describes the steps on Device R1.

Copyright © 2012, Juniper Networks, Inc.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R2

```
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor fe-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R3

```
set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure a basic RIP network:

1. Configure the network interfaces.

   This example shows multiple loopback interface addresses to simulate attached networks.

   ```
   [edit interfaces]
   user@R1# set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
   ```

```
user@R1# set interfaces lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set interfaces lo0 unit 1 family inet address 192.168.1.1/32
```

2. Create the RIP group and add the interface.

   To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```
[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

4. Apply the routing policy.

   In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
   unit 1 {
      family inet {
         address 10.0.0.1/30;
      }
   }
}
lo0 {
   unit 1 {
      family inet {
         address 172.16.0.1/32;
         address 192.168.1.1/32;
      }
   }
}

user@R1# show protocols
rip {
   group rip-group {
      export advertise-routes-through-rip;
      neighbor fe-1/2/0.1;
   }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
   term 1 {
```

 Copyright © 2012, Juniper Networks, Inc.

```
        from protocol [ direct rip ];
        then accept;
      }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Checking the Routing Table on page 141
- Looking at the Routes That Device R1 Is Advertising to Device R2 on page 141
- Looking at the Routes That Device R1 Is Receiving from Device R2 on page 142
- Verifying the RIP-Enabled Interfaces on page 142
- Verifying the Exchange of RIP Messages on page 143
- Verifying Reachability of All Hosts in the RIP Network on page 143

Checking the Routing Table

Purpose Verify that the routing table is populated with the expected routes..

Action From operational mode, enter the **show route protocol rip** command.

```
user@R1> show route protocol rip
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/30        *[RIP/100] 00:59:15, metric 2, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
172.16.2.2/32      *[RIP/100] 02:52:48, metric 2, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
172.16.3.3/32      *[RIP/100] 00:45:05, metric 3, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
192.168.2.2/32     *[RIP/100] 02:52:48, metric 2, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
192.168.3.3/32     *[RIP/100] 00:45:05, metric 3, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
224.0.0.9/32       *[RIP/100] 00:45:09, metric 1
                          MultiRecv
```

Meaning The output shows that the routes have been learned from Device R2 and Device R3.

If you were to delete the **from protocol rip** condition in the routing policy on Device R2, the remote routes from Device R3 would not be learned on Device R1.

Looking at the Routes That Device R1 Is Advertising to Device R2

Purpose Verify that Device R1 is sending the expected routes.

Action From operational mode, enter the **show route advertising-protocol rip** command.

```
user@R1> show route advertising-protocol rip 10.0.0.1
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
172.16.0.1/32      *[Direct/0] 05:18:26
                    >    via lo0.1
192.168.1.1/32     *[Direct/0] 05:18:25
                    >    via lo0.1
```

Meaning Device R1 is sending routes to its directly connected networks.

Looking at the Routes That Device R1 Is Receiving from Device R2

Purpose Verify that Device R1 is receiving the expected routes.

Action From operational mode, enter the **show route receive-protocol rip** command.

```
user@R1> show route receive-protocol rip 10.0.0.2
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/30        *[RIP/100] 02:31:22, metric 2, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
172.16.2.2/32      *[RIP/100] 04:24:55, metric 2, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
172.16.3.3/32      *[RIP/100] 02:17:12, metric 3, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
192.168.2.2/32     *[RIP/100] 04:24:55, metric 2, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
192.168.3.3/32     *[RIP/100] 02:17:12, metric 3, tag 0
                    > to 10.0.0.2 via fe-1/2/0.1
```

Meaning Device R1 is receiving from Device R2 all of Device R2's directly connected networks. Device R1 is also receiving from Device R2 all of Device R3's directly connected networks, which Device R2 learned from Device R3 through RIP.

Verifying the RIP-Enabled Interfaces

Purpose Verify that all RIP-enabled Interfaces are available and active.

Action From operational mode, enter the **show rip neighbor** command.

```
user@R1> show rip neighbor
                  Local  Source          Destination     Send   Receive   In
Neighbor          State  Address         Address         Mode   Mode     Met
--------          -----  -------         -----------     ----   -------  ---
fe-1/2/0.1           Up 10.0.0.1        224.0.0.9       mcast  both       1
```

Meaning The output shows that the RIP-enabled interface on Device R1 is operational.

In general for this command, the output shows a list of the RIP neighbors that are configured on the device. Verify the following information:

- Each configured interface is present. Interfaces are listed in alphabetical order.
- Each configured interface is up. The state of the interface is listed in the **Destination State** column. A state of **Up** indicates that the link is passing RIP traffic. A state of **Dn** indicates that the link is not passing RIP traffic. In a point-to-point link, this state

 Copyright © 2012, Juniper Networks, Inc.

generally means that either the end point is not configured for RIP or the link is unavailable.

Verifying the Exchange of RIP Messages

Purpose Verify that RIP messages are being sent and received on all RIP-enabled interfaces.

Action From operational mode, enter the **show rip statistics** command.

```
user@R1> show rip statistics
RIPv2 info: port 520; holddown 120s.
    rts learned  rts held down  rqsts dropped  resps dropped
              5              0              0              0

fe-1/2/0.1:  5 routes learned; 2 routes advertised; timeout 180s; update interval
 30s
Counter                         Total   Last 5 min  Last minute
-------                     -----------  -----------  -----------
Updates Sent                     2669           10            2
Triggered Updates Sent              2            0            0
Responses Sent                      0            0            0
Bad Messages                        0            0            0
RIPv1 Updates Received              0            0            0
RIPv1 Bad Route Entries             0            0            0
RIPv1 Updates Ignored               0            0            0
RIPv2 Updates Received           2675           11            2
RIPv2 Bad Route Entries             0            0            0
RIPv2 Updates Ignored               0            0            0
Authentication Failures             0            0            0
RIP Requests Received               0            0            0
RIP Requests Ignored                0            0            0
none                                0            0            0
```

Meaning The output shows the number of RIP routes learned. It also shows the number of RIP updates sent and received on the RIP-enabled interfaces. Verify the following information:

- The number of RIP routes learned matches the number of expected routes learned. Subnets learned by direct connectivity through an outgoing interface are not listed as RIP routes.
- RIP updates are being sent on each RIP-enabled interface. If no updates are being sent, the routing policy might not be configured to export routes.
- RIP updates are being received on each RIP-enabled interface. If no updates are being received, the routing policy might not be configured to export routes on the host connected to that subnet. The lack of updates might also indicate an authentication error.

Verifying Reachability of All Hosts in the RIP Network

Purpose By using the traceroute command on each loopback address in the network, verify that all hosts in the RIP network are reachable from each Juniper Networks device.

Action From operational mode, enter the **traceroute** command.

```
user@R1> traceroute 192.168.3.3
```

```
traceroute to 192.168.3.3 (192.168.3.3), 30 hops max, 40 byte packets
 1  10.0.0.2 (10.0.0.2)  1.094 ms  1.028 ms  0.957 ms
 2  192.168.3.3 (192.168.3.3)  1.344 ms  2.245 ms  2.125 ms
```

Meaning Each numbered row in the output indicates a routing hop in the path to the host. The three-time increments indicate the round-trip time (RTT) between the device and the hop for each traceroute packet.

To ensure that the RIP network is healthy, verify the following information:

- The final hop in the list is the host you want to reach.
- The number of expected hops to the host matches the number of hops in the traceroute output. The appearance of more hops than expected in the output indicates that a network segment is probably unreachable. It might also indicate that the incoming or outgoing metric on one or more hosts has been set unexpectedly.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Basic RIP Routing on page 137
- RIP Configuration Overview on page 137

RIP Traffic Control Through Metrics

- Understanding RIP Traffic Control with Metrics on page 144
- Example: Controlling Traffic with an Incoming Metric on page 145
- Example: Controlling Traffic with an Outgoing Metric on page 146

Understanding RIP Traffic Control with Metrics

To tune a RIP network and control traffic flowing through the network, you increase or decrease the cost of the paths through the network. RIP provides two ways to modify the path cost: an incoming metric and an outgoing metric, which are each set to **1** by default. These metrics are attributes that manually specify the cost of any route advertised through a host. By increasing or decreasing the metrics—and thus the cost—of links throughout the network, you can control packet transmission across the network.

The incoming metric modifies the cost of an individual segment when a route across the segment is imported into the routing table. For example, if you set the incoming metric on the segment to **3**, the individual segment cost along the link is changed from **1** to **3**. The increased cost affects all route calculations through that link. Other routes that were previously excluded because of a high hop count might now be selected into the router's forwarding table.

The outgoing metric modifies the path cost for all the routes advertised out a particular interface. Unlike the incoming metric, the outgoing metric modifies the routes that other routers are learning and thereby controls the way they send traffic.

If an exported route was learned from a member of the same RIP group, the metric associated with that route is the normal RIP metric. For example, a RIP route with a metric

 Copyright © 2012, Juniper Networks, Inc.

of 5 learned from a neighbor configured with an incoming metric of 2 is advertised with a combined metric of 7 when advertised to neighbors in the same group. However, if this route was learned from a RIP neighbor in a different group or from a different protocol, the route is advertised with the metric value configured in the outgoing metric for that group.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- RIP Overview on page 131
- Example: Controlling Traffic with an Incoming Metric on page 145
- Example: Controlling Traffic with an Outgoing Metric on page 146

Example: Controlling Traffic with an Incoming Metric

This example shows how to control traffic with an incoming metric.

- Requirements on page 145
- Overview on page 145
- Configuration on page 146
- Verification on page 146

Requirements

Before you begin, define RIP groups, and add interfaces to the groups. Then configure a routing policy to export directly connected routes and routes learned through the RIP routing exchanges. See "Example: Configuring a Basic RIP Network" on page 138.

Overview

In this example, routes to Router D are received by Router A across both of its RIP-enabled interfaces as shown in Figure 24 on page 145. Because the route through Router B and the route through Router C have the same number of hops, both routes are imported into the forwarding table. However, because the T3 link from Router B to Router D has a higher bandwidth than the T1 link from Router C to Router D, you want traffic to flow from A through B to D.

Figure 24: Controlling Traffic in a RIP Network with the Incoming Metric

To force this flow, you can modify the route metrics as they are imported into Router A's routing table. By setting the incoming metric on the interface from Router A to Router C,

you modify the metric on all routes received through that interface. Setting the incoming route metric on Router A changes only the routes in Router A's routing table, and affects only how Router A sends traffic to Router D. Router D's route selection is based on its own routing table, which, by default, includes no adjusted metric values.

In the example, Router C receives a route advertisement from Router D and readvertises the route to Router A. When Router A receives the route, it applies the incoming metric on the interface. Instead of incrementing the metric by 1 (the default), Router A increments it by 3 (the configured incoming metric), giving the route from Router A to Router D through Router C a total path metric of 4. Because the route through Router B has a metric of 2, it becomes the preferred route for all traffic from Router A to Router D.

This example uses a RIP group called **alpha 1** on interface **g3–0/0/0**.

Configuration

Step-by-Step Procedure

To control traffic with an incoming metric:

1. Create an interface.

```
[edit]
user@host# edit protocols rip group alpha1 neighbor ge-0/0/0
```

2. Set the incoming metric.

```
[edit]
user@host# set metric-in 3
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show protocols rip** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding RIP Traffic Control with Metrics on page 144 on page 29
- RIP Configuration Overview on page 137
- Example: Controlling Traffic with an Outgoing Metric on page 146
- Verifying a RIP Configuration on page 150

Example: Controlling Traffic with an Outgoing Metric

This example shows how to control traffic with an outgoing metric.

- Requirements on page 147
- Overview on page 147
- Configuration on page 147
- Verification on page 148

Copyright © 2012, Juniper Networks, Inc.

Requirements

Before you begin:

- Define RIP groups, and add interfaces to the groups. Then configure a routing policy to export directly connected routes and routes learned through RIP routing exchanges. See "Example: Configuring a Basic RIP Network" on page 138.
- Control traffic with an incoming metric. See "Example: Controlling Traffic with an Incoming Metric" on page 145.

Overview

In this example, each route from Router A to Router D has two hops as shown in Figure 25 on page 147. However, because the link from Router A to Router B in RIP group Beta 1 has a higher bandwidth than the link from Router A to Router C in RIP group Alpha 1, you want traffic from Router D to Router A to flow through Router B. To control the way Router D sends traffic to Router A, you can alter the routes that Router D receives by configuring the outgoing metric on Router A's interfaces in the Alpha 1 RIP group.

Figure 25: Controlling Traffic in a RIP Network with the Outgoing Metric

If the outgoing metric for the Alpha 1 RIP group—the A-to-C link—is changed to 3, Router D calculates the total path metric from A through C as 4. In contrast, the unchanged default total path metric to A through B in the Beta 1 RIP group is 2. The fact that Router A's interfaces belong to two different RIP groups allows you to configure two different outgoing metrics on its interfaces, because you configure path metrics at the group level.

By configuring the outgoing metric, you control the way Router A sends traffic to Router D. By configuring the outgoing metric on the same router, you control the way Router D sends traffic to Router A.

This example uses an outgoing metric of **3**.

Configuration

Step-by-Step Procedure

To control traffic with an outgoing metric:

1. Create an interface.

```
[edit]
user@host# edit protocols rip group alpha1
```

2. Set the outgoing metric.

```
[edit]
user@host# set metric-out 3
```

3. If you are done configuring the device, commit the configuration.

```
[edit ]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show protocols rip** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding RIP Traffic Control with Metrics on page 144
- RIP Configuration Overview on page 137
- Verifying a RIP Configuration on page 150

RIP Authentication

- Understanding RIP Authentication on page 148
- Enabling Authentication with Plain-Text Passwords (CLI Procedure) on page 149
- Enabling Authentication with MD5 Authentication (CLI Procedure) on page 149

Understanding RIP Authentication

RIPv2 provides authentication support so that RIP links can require authentication keys (passwords) before they become active. Authentication provides an additional layer of security on the network beyond the other security features. By default, this authentication is disabled.

Authentication keys can be specified in either plain-text or MD5 form. Authentication requires all routers within the RIP network or subnetwork to have the same authentication type and key (password) configured.

This type of authentication is not supported on RIPv1 networks.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- RIP Overview on page 131
- Enabling Authentication with Plain-Text Passwords (CLI Procedure) on page 149
- Enabling Authentication with MD5 Authentication (CLI Procedure) on page 149

Copyright © 2012, Juniper Networks, Inc.

Enabling Authentication with Plain-Text Passwords (CLI Procedure)

To configure authentication that requires a plain-text password to be included in the transmitted packet, enable simple authentication by performing these steps on all RIP devices in the network:

1. Navigate to the top of the configuration hierarchy.
2. Perform the configuration tasks described in Table 3 on page 149.
3. If you are finished configuring the router, commit the configuration.

Table 3: Configuring Simple RIP Authentication

Task	CLI Configuration Editor
Navigate to **Rip** level in the configuration hierarchy.	From the **[edit]** hierarchy level, enter **edit protocols rip**
Set the authentication type to **simple**.	Set the authentication type to **simple**: **set authentication-type simple**
Set the authentication key to a simple-text password. The password can be from 1 through 16 contiguous characters long and can include any ASCII strings.	Set the authentication key to a simple-text password: **set authentication-key *password***

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding RIP Authentication on page 148
- RIP Configuration Overview on page 137
- Enabling Authentication with MD5 Authentication (CLI Procedure) on page 149

Enabling Authentication with MD5 Authentication (CLI Procedure)

To configure authentication that requires an MD5 password to be included in the transmitted packet, enable MD5 authentication by performing these steps on all RIP devices in the network:

1. Navigate to the top of the configuration hierarchy.
2. Perform the configuration tasks described in Table 4 on page 150.
3. If you are finished configuring the router, commit the configuration.

Copyright © 2012, Juniper Networks, Inc.

Table 4: Configuring MD5 RIP Authentication

Task	CLI Configuration Editor
Navigate to **Rip** level in the configuration hierarchy.	From the **[edit]** hierarchy level, enter **edit protocols rip**
Set the authentication type to **MD5**.	Set the authentication type to **md5**: **set authentication-type md5**
Set the MD5 authentication key (password). The key can be from 1 through 16 contiguous characters long and can include any ASCII strings.	Set the MD5 authentication key: **set authentication-key *password***

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding RIP Authentication on page 148
- RIP Configuration Overview on page 137
- Enabling Authentication with Plain-Text Passwords (CLI Procedure) on page 149

Verifying a RIP Configuration

To verify a RIP configuration, perform the following tasks:

- Verifying the Exchange of RIP Messages on page 150
- Verifying the RIP-Enabled Interfaces on page 151
- Verifying Reachability of All Hosts in the RIP Network on page 152

Verifying the Exchange of RIP Messages

Purpose Verify that RIP messages are being sent and received on all RIP-enabled interfaces.

Action From the CLI, enter the **show rip statistics** command.

Sample Output

```
user@host> show rip statistics
RIPv2 info: port 520; holddown 120s.
    rts learned  rts held down  rqsts dropped  resps dropped
             10              0              0              0

t1-0/0/2.0:  0 routes learned; 13 routes advertised; timeout 120s; update interval
 45s
Counter                        Total   Last 5 min  Last minute
-------                   -----------  -----------  -----------
Updates Sent                     2855           11            2
Triggered Updates Sent              5            0            0
Responses Sent                      0            0            0
Bad Messages                        0            0            0
RIPv1 Updates Received              0            0            0
```


Copyright © 2012, Juniper Networks, Inc.

```
RIPv1 Bad Route Entries             0            0            0
RIPv1 Updates Ignored               0            0            0
RIPv2 Updates Received             41            0            0
RIPv2 Bad Route Entries             0            0            0
RIPv2 Updates Ignored               0            0            0
Authentication Failures             0            0            0
RIP Requests Received               0            0            0
RIP Requests Ignored                0            0            0

ge-0/0/1.0:  10 routes learned; 3 routes advertised; timeout 180s; update interval
 30s
Counter                        Total   Last 5 min  Last minute
-------                   -----------  -----------  -----------
Updates Sent                     2855           11            2
Triggered Updates Sent              3            0            0
Responses Sent                      0            0            0
Bad Messages                        1            0            0
RIPv1 Updates Received              0            0            0
RIPv1 Bad Route Entries             0            0            0
RIPv1 Updates Ignored               0            0            0
RIPv2 Updates Received           2864           11            2
RIPv2 Bad Route Entries            14            0            0
RIPv2 Updates Ignored               0            0            0
Authentication Failures             0            0            0
RIP Requests Received               0            0            0
RIP Requests Ignored                0            0            0
```

Meaning The output shows the number of RIP routes learned. It also shows the number of RIP updates sent and received on the RIP-enabled interfaces. Verify the following information:

- The number of RIP routes learned matches the number of expected routes learned. Subnets learned by direct connectivity through an outgoing interface are not listed as RIP routes.
- RIP updates are being sent on each RIP-enabled interface. If no updates are being sent, the routing policy might not be configured to export routes.
- RIP updates are being received on each RIP-enabled interface. If no updates are being received, the routing policy might not be configured to export routes on the host connected to that subnet. The lack of updates might also indicate an authentication error.

Verifying the RIP-Enabled Interfaces

Purpose Verify that all the RIP-enabled interfaces are available and active.

Action From the CLI, enter the **show rip neighbor** command.

Sample Output

```
user@host> show rip neighbor
Source          Destination     Send   Receive   In
Neighbor          State  Address         Address         Mode   Mode     Met
--------          -----  -------         -----------     ----   -------  ---
ge-0/0/0.0           Dn (null)          (null)          mcast  both       1
ge-0/0/1.0           Up 192.168.220.5   224.0.0.9       mcast  both       1
```

Meaning The output shows a list of the RIP neighbors that are configured on the device. Verify the following information:

- Each configured interface is present. Interfaces are listed in alphabetical order.
- Each configured interface is up. The state of the interface is listed in the **Destination State** column. A state of **Up** indicates that the link is passing RIP traffic. A state of **Dn** indicates that the link is not passing RIP traffic. In a point-to-point link, this state generally means that either the end point is not configured for RIP or the link is unavailable.

Verifying Reachability of All Hosts in the RIP Network

Purpose By using the traceroute tool on each loopback address in the network, verify that all hosts in the RIP network are reachable from each Juniper Networks device.

Action For each device in the RIP network:

1. In the J-Web interface, select **Troubleshoot>Traceroute**.
2. In the Remote Host box, type the name of a host for which you want to verify reachability from the device.
3. Click **Start**. Output appears on a separate page.

Sample Output

```
1 172.17.40.254 (172.17.40.254) 0.362 ms 0.284 ms 0.251 ms
2 routera-fxp0.englab.mycompany.net (192.168.71.246) 0.251 ms 0.235 ms 0.200 ms
```

Meaning Each numbered row in the output indicates a routing hop in the path to the host. The three-time increments indicate the round-trip time (RTT) between the device and the hop for each traceroute packet.

To ensure that the RIP network is healthy, verify the following information:

- The final hop in the list is the host you want to reach.
- The number of expected hops to the host matches the number of hops in the traceroute output. The appearance of more hops than expected in the output indicates that a network segment is probably unreachable. It might also indicate that the incoming or outgoing metric on one or more hosts has been set unexpectedly.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- RIP Configuration Overview on page 137
- show rip statistics in the *Junos OS Routing Protocols and Policies Command Reference*
- show rip neighbor in the *Junos OS Routing Protocols and Policies Command Reference*
- traceroute in the *Junos OS System Basics and Services Command Reference*
- RIP Overview on page 131

 Copyright © 2012, Juniper Networks, Inc.

RIP Demand Circuits Overview

RIP periodically sends routing information (RIP packets) to neighboring devices. These periodic broadcasts can consume bandwidth resources and interfere with network traffic by preventing WAN circuits from being closed. Demand circuits for RIP is defined in RFC 2091 and overcomes these issues by exchanging incremental updates on demand.

A demand circuit is a point-to-point connection between two neighboring interfaces configured for RIP. Demand circuits preserve bandwidth by establishing a link when data needs to be transferred, and terminating the link when the data transfer is complete. Demand circuits increase the efficiency of RIP on the configured interfaces by offering minimal network overhead in terms of messages passed between the demand circuit end points, conserving resources, and reducing costs.

By configuring RIP demand circuits, a specific event triggers the device to send an update, thereby eliminating the periodic transmission of RIP packets over the neighboring interface. To save overhead, the device sends RIP information only when changes occur in the routing database, such as:

- The device is first powered on
- The device receives a request for route update information
- A change occurs in the network
- The demand circuit goes down or comes up

The device sends update requests, update responses, and acknowledgments. In addition, the device retransmits updates and requests until valid acknowledgments are received. The device dynamically learns RIP neighbors. If the neighboring interface goes down, RIP flushes routes learned from the neighbor's IP address.

Routes learned from demand circuits do not age like other RIP entries because demand circuits are in a permanent state. Routes in a permanent state are only removed under the following conditions:

- A formerly reachable route changes to unreachable in an incoming response
- The demand circuit is down due to an excessive number of unacknowledged retransmissions

You can also set the RIP hold-down timer and the RIP demand circuit retransmission timer to regulate performance. The demand circuit uses these timers to determine if there is a change that requires update messages to be sent. There is also a database timer that runs only when RIP flushes learned routes from the routing table.

This topic includes the following sections:

- RIP Demand Circuit Packets on page 154
- Timers Used by RIP Demand Circuits on page 154

Copyright © 2012, Juniper Networks, Inc.

RIP Demand Circuit Packets

When you configure an interface for RIP demand circuits, the supported command field packet types are different than those for RIP version 1 and RIP version 2. RIP packets for RIP demand circuits contain three additional packet types and an extended 4-byte update header. Both RIP version 1 and RIP version 2 support the three packet types and the extended 4-byte header. Table 5 on page 154 describes the three packet types.

Table 5: RIP Demand Circuit Packet Types

Packet Type	Description
Update Request	Update request messages seek information for the device's routing table. This message is sent when the device is first powered on or when a down demand circuit comes up. The device sends this message every 5 seconds (by default) until an update response message is received.
Update Response	Update response messages are sent in response to an update request message, which occurs when the device is first powered on or when a down demand circuit comes up. Each update response message contains a sequence number that the neighbor uses to acknowledge the update request.
Update Acknowledge	Update acknowledge messages are sent in response to every update response message received by the neighbor.

NOTE: **These packets are only valid on interfaces configured for RIP demand circuits. If a demand circuit receives a RIP packet that does not contain these packet types, it silently discards the packet and logs an error message similar to the following:**

Ignoring RIP packet with invalid version 0 from neighbor 10.0.0.0 and source 10.0.0.1

Related Documentation

- RIP Overview
- demand-circuit

Timers Used by RIP Demand Circuits

RIP demand circuits use the RIP hold-down timer and the RIP demand circuit retransmission timer to regulate performance and to determine if there is a change in the network that requires the device to send update messages. The hold-down timer is a global RIP timer that affects the entire RIP configuration; whatever range you configure for RIP applies to RIP demand circuits. The retransmission timer affects only RIP demand circuits. In addition, there is a database timer that runs only when RIP flushes learned routes from the routing table.

- Hold-down timer (global RIP timer)—Use the hold-down timer to configure the number of seconds that RIP waits before updating the routing table. The value of the hold-down

 Copyright © 2012, Juniper Networks, Inc.

timer affects the entire RIP configuration, not just the demand circuit interfaces. The hold-down timer starts when a route timeout limit is met, when a formerly reachable route is unreachable, or when a demand circuit interface is down. When the hold-down timer is running, routes are advertised as unreachable on other interfaces. When the hold-down timer expires, the route is removed from the routing table if all destinations are aware that the route is unreachable or the remaining destinations are down. By default, RIP waits 120 seconds between routing table updates. The range is from 10 to 180 seconds.

- Retransmission timer (RIP demand circuit timer)—RIP demand circuits send update messages every 5 seconds to an unresponsive peer. Use the retransmission timer to limit the number of times a demand circuit resends update messages to an unresponsive peer. If the configured retransmission threshold is reached, routes from the next hop router are marked as unreachable and the hold-down timer starts. The value of the retransmission timer affects only the demand circuit interfaces. To determine the number of times to resend the update message, use the following calculation:

  5 seconds * number of retransmissions = retransmission seconds

  The retransmission range is from 5 through 180 seconds, which corresponds to sending an update message a minimum of 1 time (5 seconds) and a maximum of 36 times (180 seconds).

- Database timer (global timeout timer)—Routes learned from demand circuits do not age like other RIP entries because demand circuits are in a permanent state. On a RIP demand circuit, the database timer starts upon receipt of the update response message with the flush flag sent from a RIP demand circuit peer. When the neighbor receives this message, all routes from that peer are flushed, and the database timer starts and runs for the configured route timeout interval. When the database timer is running, routes are still advertised as reachable on other interfaces. When the database timer expires, the device advertises all routes from its peer as unreachable.

Related Documentation

- Configuring RIP Timers
- Example: Configuring RIP Demand Circuits on page 155
- holddown
- max-retrans-time

Example: Configuring RIP Demand Circuits

This example describes how to configure an interface as a RIP demand circuit.

- Requirements on page 156
- Overview on page 156
- Configuration on page 156
- Verification on page 158

Requirements

Before you begin, configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.

Overview

A demand circuit is a point-to-point connection between two neighboring interfaces configured for RIP. Demand circuits increase the efficiency of RIP on the configured interfaces by eliminating the periodic transmission of RIP packets. Demand circuits preserve bandwidth by establishing a link when data needs to be transferred, and terminating the link when the data transfer is complete. In this example two devices are connected using SONET/SDH interfaces.

NOTE: **When you configure RIP demand circuits, any silent removal of the RIP configuration goes unnoticed by the RIP peer and leads to stale entries in the routing table. To clear the stale entries, deactivate and reactivate RIP on the neighboring devices.**

In this example, you configure interface **so-0/1/0** with the following settings:

- **demand-circuit**—Configures the interface as a demand circuit. To complete the demand circuit, you must configure both ends of the pair as demand circuits.

- **max-retrans-time**—RIP demand circuits send update messages every 5 seconds to an unresponsive peer. Use the retransmission timer to limit the number of times a demand circuit resends update messages to an unresponsive peer. If the configured retransmission threshold is reached, routes from the next-hop router are marked as unreachable, and the hold-down timer starts. The value of the retransmission timer affects only the demand circuit interfaces. To determine the number of times to resend the update message, use the following calculation:

  5 seconds * retransmissions = retransmission seconds

  For example, if you want the demand circuit to send only two update messages to an unresponsive peer, the calculation is: 5 * 2 = 10. When you configure the retransmission timer, you enter 10 seconds.

  The retransmission range is from 5 through 180 seconds, which corresponds to sending an update message a minimum of 1 time (5 seconds) and a maximum of 36 times (180 seconds).

Configuration

In the following example, you configure a neighboring interface to be a RIP demand circuit and save the configuration.

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network

Copyright © 2012, Juniper Networks, Inc.

configuration, and then copy and paste the commands in the CLI at the **[edit]** hierarchy level.

```
set interfaces so-0/1/0 unit 0 family inet address 192.0.2.0/24
set protocols rip group group1 neighbor so-0/1/0 demand-circuit
set protocols rip group group1 neighbor so-0/1/0 max-retrans-time 10
```

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure a RIP demand circuit on one neighboring interface:

1. Configure the interface.

   ```
   [edit]
   user@host# set interfaces so-0/1/0 unit 0 family inet address 192.0.2.0/24
   ```

2. Enter RIP configuration mode.

   ```
   [edit]
   user@host# edit protocols rip
   ```

3. Configure the neighbor as a demand circuit.

   ```
   [edit protocols rip]
   user@host# set group group1 neighbor so-0/1/0 demand-circuit
   ```

4. Configure the demand circuit retransmission timer.

   ```
   [edit protocols rip]
   user@host# set group group1 neighbor so-0/1/0 max-retrans-time 10
   ```

5. If you are done configuring the device, commit the configuration.

   ```
   [edit protocols rip]
   user@host# commit
   ```

NOTE: Repeat this entire configuration on the other neighboring interface.

Results

Confirm your configuration by entering the **show interfaces** and **show protocols rip** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show interfaces
so-0/1/0 {
   unit 0 {
      family inet {
         address 192.0.2.0/24;
      }
   }
}

user@host# show protocols rip
```

```
group group1 {
  neighbor so-0/1/0 {
    demand-circuit;
    max-retrans-time 10;
  }
}
```

Verification

Verifying a Demand Circuit Configuration

Purpose Verify that the demand circuit configuration is working.

Action To verify that the demand circuit configuration is in effect, use the **show rip neighbor** operational mode command.

```
user@host# show rip neighbor
                          Source          Destination     Send   Receive  In
Neighbor          State   Address         Address         Mode   Mode     Met
--------          -----   -------         -----------     ----   -------  ---
so-0/1/0.0(DC)       Up   10.10.10.2       224.0.0.9      mcast  both       1
```

When you configure demand circuits, the **show rip neighbor** command displays a DC flag next to the neighboring interface configured for demand circuits.

NOTE: If you configure demand circuits at the **[edit protocols rip group *group-name* neighbor *neighbor-name*]** hierarchy level, the output shows only the neighboring interface that you specifically configured as a demand circuit. If you configure demand circuits at the **[edit protocols rip group *group-name*]** hierarchy level, all of the interfaces in the group are configured as demand circuits. Therefore, the output shows all of the interfaces in that group as demand circuits.

Related Documentation

- RIP Demand Circuits Overview on page 153

 Copyright © 2012, Juniper Networks, Inc.

CHAPTER 9

OSPF

- OSPF Overview on page 160
- OSPF Configuration Overview on page 165
- OSPF Designated Routers on page 166
- OSPF Areas, Area Border Routers, and Backbone Areas on page 170
- OSPF Stub Areas and Not-So-Stubby Areas on page 176
- OSPF Authentication on page 187
- OSPF Traffic Control on page 202
- Verifying an OSPF Configuration on page 210

Copyright © 2012, Juniper Networks, Inc.

OSPF Overview

Copyright © 2012, Juniper Networks, Inc.

OSPF is an interior gateway protocol (IGP) that routes packets within a single autonomous system (AS). OSPF uses link-state information to make routing decisions, making route calculations using the shortest-path-first (SPF) algorithm (also referred to as the Dijkstra algorithm). Each router running OSPF floods link-state advertisements throughout the AS or area that contain information about that router's attached interfaces and routing metrics. Each router uses the information in these link-state advertisements to calculate the least cost path to each network and create a routing table for the protocol.

Junos OS supports OSPF version 2 (OSPFv2) and OSPF version 3 (OSPFv3), including virtual links, stub areas, and for OSPFv2, authentication. Junos OS does not support type-of-service (ToS) routing.

OSPF was designed for the Transmission Control Protocol/Internet Protocol (TCP/IP) environment and as a result explicitly supports IP subnetting and the tagging of externally derived routing information. OSPF also provides for the authentication of routing updates.

OSPF routes IP packets based solely on the destination IP address contained in the IP packet header. OSPF quickly detects topological changes, such as when router interfaces become unavailable, and calculates new loop-free routes quickly and with a minimum of routing overhead traffic.

NOTE: **On SRX Series devices, when only one link-protection is configured under the OSPF interface, the device does not install an alternative route in the forwarding table. When the per-packet load-balancing is enabled as a workaround, the device does not observe both the OSPF metric and sending the traffic through both the interfaces.**

An OSPF AS can consist of a single area, or it can be subdivided into multiple areas. In a single-area OSPF network topology, each router maintains a database that describes the topology of the AS. Link-state information for each router is flooded throughout the AS. In a multiarea OSPF topology, each router maintains a database that describes the topology of its area, and link-state information for each router is flooded throughout that area. All routers maintain summarized topologies of other areas within an AS. Within each area, OSPF routers have identical topological databases. When the AS or area topology changes, OSPF ensures that the contents of all routers' topological databases converge quickly.

All OSPFv2 protocol exchanges can be authenticated. OSPFv3 relies on IPsec to provide this functionality. This means that only trusted routers can participate in the AS's routing. A variety of authentication schemes can be used. A single authentication scheme is configured for each area, which enables some areas to use stricter authentication than others.

Externally derived routing data (for example, routes learned from BGP) is passed transparently throughout the AS. This externally derived data is kept separate from the OSPF link-state data. Each external route can be tagged by the advertising router, enabling the passing of additional information between routers on the boundaries of the AS.

Copyright © 2012, Juniper Networks, Inc.

NOTE: By default, Junos OS is compatible with RFC 1583, *OSPF Version 2*. In Junos OS Release 8.5 and later, you can disable compatibility with RFC 1583 by including the **no-rfc-1583** statement. For more information, see Example: Disabling OSPFv2 Compatibility with RFC 1583.

This topic describes the following information:

- OSPF Default Route Preference Values on page 162
- OSPF Routing Algorithm on page 162
- OSPF Three-Way Handshake on page 163
- OSPF Version 3 on page 164

OSPF Default Route Preference Values

The Junos OS routing protocol process assigns a default preference value to each route that the routing table receives. The default value depends on the source of the route. The preference value is from 0 through 4,294,967,295 (232 – 1), with a lower value indicating a more preferred route. Table 6 on page 162 lists the default preference values for OSPF.

Table 6: Default Route Preference Values for OSPF

How Route Is Learned	Default Preference	Statement to Modify Default Preference
OSPF internal route	10	OSPF preference
OSPF AS external routes	150	OSPF external-preference

OSPF Routing Algorithm

OSPF uses the shortest-path-first (SPF) algorithm, also referred to as the Dijkstra algorithm, to determine the route to each destination. All routing devices in an area run this algorithm in parallel, storing the results in their individual topological databases. Routing devices with interfaces to multiple areas run multiple copies of the algorithm. This section provides a brief summary of how the SPF algorithm works.

When a routing device starts, it initializes OSPF and waits for indications from lower-level protocols that the router interfaces are functional. The routing device then uses the OSPF hello protocol to acquire neighbors, by sending hello packets to its neighbors and receiving their hello packets.

On broadcast or nonbroadcast multiaccess networks (physical networks that support the attachment of more than two routing devices), the OSPF hello protocol elects a designated router for the network. This routing device is responsible for sending *link-state advertisements* (LSAs) that describe the network, which reduces the amount of network traffic and the size of the routing devices' topological databases.

The routing device then attempts to form *adjacencies* with some of its newly acquired neighbors. (On multiaccess networks, only the designated router and backup designated

Copyright © 2012, Juniper Networks, Inc.

router form adjacencies with other routing devices.) Adjacencies determine the distribution of routing protocol packets. Routing protocol packets are sent and received only on adjacencies, and topological database updates are sent only along adjacencies. When adjacencies have been established, pairs of adjacent routers synchronize their topological databases.

A routing device sends LSA packets to advertise its state periodically and when its state changes. These packets include information about the routing device's adjacencies, which allows detection of nonoperational routing devices.

Using a reliable algorithm, the routing device floods LSAs throughout the area, which ensures that all routing devices in an area have exactly the same topological database. Each routing device uses the information in its topological database to calculate a shortest-path tree, with itself as the root. The routing device then uses this tree to route network traffic.

The description of the SPF algorithm up to this point has explained how the algorithm works within a single area (*intra-area routing*). For internal routers to be able to route to destinations outside the area (*interarea routing*), the area border routers must inject additional routing information into the area. Because the area border routers are connected to the backbone, they have access to complete topological data about the backbone. The area border routers use this information to calculate paths to all destinations outside its area and then advertise these paths to the area's internal routers.

Autonomous system (AS) boundary routers flood information about external autonomous systems throughout the AS, except to stub areas. Area border routers are responsible for advertising the paths to all AS boundary routers.

OSPF Three-Way Handshake

OSPF creates a topology map by flooding LSAs across OSPF-enabled links. LSAs announce the presence of OSPF-enabled interfaces to adjacent OSPF interfaces. The exchange of LSAs establishes bidirectional connectivity between all adjacent OSPF interfaces (neighbors) using a three-way handshake, as shown in Figure 26 on page 163.

Figure 26: OSPF Three-Way Handshake

In Figure 26 on page 163, Router A sends hello packets out all its OSPF-enabled interfaces when it comes online. Router B receives the packet, which establishes that Router B can receive traffic from Router A. Router B generates a response to Router A to acknowledge receipt of the hello packet. When Router A receives the response, it establishes that Router B can receive traffic from Router A. Router A then generates a final response packet to inform Router B that Router A can receive traffic from Router B. This three-way handshake ensures bidirectional connectivity.

Copyright © 2012, Juniper Networks, Inc.

As new neighbors are added to the network or existing neighbors lose connectivity, the adjacencies in the topology map are modified accordingly through the exchange (or absence) of LSAs. These LSAs advertise only the incremental changes in the network, which helps minimize the amount of OSPF traffic on the network. The adjacencies are shared and used to create the network topology in the topological database.

OSPF Version 3

OSPFv3 is a modified version of OSPF that supports IP version 6 (IPv6) addressing. OSPFv3 differs from OSPFv2 in the following ways:

- All neighbor ID information is based on a 32-bit router ID.
- The protocol runs per link rather than per subnet.
- Router and network link-state advertisements (LSAs) do not carry prefix information.
- Two new LSA types are included: link-LSA and intra-area-prefix-LSA.
- Flooding scopes are as follows:
  - Link-local
  - Area
  - AS
- Link-local addresses are used for all neighbor exchanges except virtual links.
- Authentication is removed. The IPv6 authentication header relies on the IP layer.
- The packet format has changed as follows:
  - Version number 2 is now version number 3.
  - The **db** option field has been expanded to 24 bits.
  - Authentication information has been removed.
  - Hello messages do not have address information.
  - Two new option bits are included: **R** and **V6**.
- Type 3 summary LSAs have been renamed *inter-area-prefix-LSAs*.
- Type 4 summary LSAs have been renamed *inter-area-router-LSAs*.

Related Documentation

- Understanding OSPF Areas and Backbone Areas on page 170
- OSPF Configuration Overview on page 165
- *Junos OS OSPF Version 3 for IPv6 Feature Guide*
- Example: Disabling OSPFv2 Compatibility with RFC 1583

Copyright © 2012, Juniper Networks, Inc.

OSPF Configuration Overview

To activate OSPF on a network, you must enable the protocol on all interfaces within the network on which OSPF traffic is to travel. To enable OSPF, you must configure one or more interfaces on the device within an OSPF area. Once the interfaces are configured, OSPF link-state advertisements (LSAs) are transmitted on all OSPF-enabled interfaces, and the network topology is shared throughout the network.

To complete the minimum device configuration for a node in an OSPF network involves:

1. Configuring the device interfaces.
   See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Configuring the router identifiers for the devices in your OSPF network
3. Creating the backbone area (area 0) for your OSPF network and adding the appropriate interfaces to the area

NOTE: **Once you complete this step, OSPF begins sending LSAs. No additional configuration is required to enable OSPF traffic on the network.**

You can further define your OSPF network depending on your network requirements. Some optional configurations involve:

- Adding additional areas to your network and configure area border routers (ABRs)
- Enabling dial-on-demand routing backup on the OSPF-enabled interface to configure OSPF across a demand circuit such as an ISDN link. (You must have already configured an ISDN interface.) Because demand circuits do not pass all traffic required to maintain an OSPF adjacency (hello packets, for example), you configure dial-on-demand routing so individual nodes in an OSPF network can maintain adjacencies despite the lack of LSA exchanges.
- Reducing the amount of memory that the nodes use to maintain the topology database by configuring stub and not-so-stubby areas
- Ensuring that only trusted routing devices participate in the autonomous systems' routing by enabling authentication
- Controlling the flow of traffic across the network by configuring path metrics and route selection

When describing how to configure OSPF, the following terms are used as follows:

- OSPF refers to both OSPF version 2 (OSPFv2) and OSPF version 3 (OSPFv3)
- OSPFv2 refers to OSPF version 2
- OSPFv3 refers to OSPF version 3

Copyright © 2012, Juniper Networks, Inc.

OSPF Designated Routers

- OSPF Designated Router Overview on page 166
- Example: Configuring an OSPF Router Identifier on page 167
- Example: Controlling OSPF Designated Router Election on page 168

OSPF Designated Router Overview

Large LANs that have many routing devices and therefore many OSPF adjacencies can produce heavy control-packet traffic as link-state advertisements (LSAs) are flooded across the network. To alleviate the potential traffic problem, OSPF uses designated routers on all multiaccess networks (broadcast and nonbroadcast multiaccess [NBMA] networks types). Rather than broadcasting LSAs to all their OSPF neighbors, the routing devices send their LSAs to the designated router. Each multiaccess network has a designated router, which performs two main functions:

- Originate network link advertisements on behalf of the network.
- Establish adjacencies with all routing devices on the network, thus participating in the synchronizing of the link-state databases.

In LANs, the election of the designated router takes place when the OSPF network is initially established. When the first OSPF links are active, the routing device with the highest router identifier (defined by the **router-id** configuration value, which is typically the IP address of the routing device, or the loopback address) is elected the designated router. The routing device with the second highest router identifier is elected the backup designated router. If the designated router fails or loses connectivity, the backup designated router assumes its role and a new backup designated router election takes place between all the routers in the OSPF network.

OSPF uses the router identifier for two main purposes: to elect a designated router, unless you manually specify a priority value, and to identify the routing device from which a packet is originated. At designated router election, the router priorities are evaluated first, and the routing device with the highest priority is elected designated router. If router priorities tie, the routing device with the highest router identifier, which is typically the routing device's IP address, is chosen as the designated router. If you do not configure a router identifier, the IP address of the first interface to come online is used. This is usually the loopback interface. Otherwise, the first hardware interface with an IP address is used.

At least one routing device on each logical IP network or subnet must be eligible to be the designated router for OSPFv2. At least one routing device on each logical link must be eligible to be the designated router for OSPFv3.

By default, routing devices have a priority of 128. A priority of 0 marks the routing device as ineligible to become the designated router. A priority of 1 means the routing device has the least chance of becoming a designated router. A priority of 255 means the routing device is always the designated router.

Related Documentation

- Example: Configuring an OSPF Router Identifier on page 167

Copyright © 2012, Juniper Networks, Inc.

- Example: Controlling OSPF Designated Router Election on page 168

Example: Configuring an OSPF Router Identifier

This example shows how to configure an OSPF router identifier.

- Requirements on page 167
- Overview on page 167
- Configuration on page 167
- Verification on page 168

Requirements

Before you begin:

- Identify the interfaces on the routing device that will participate in OSPF. You must enable OSPF on all interfaces within the network on which OSPF traffic is to travel.
- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.

Overview

In this example, you configure the OSPF router identifier by setting its router ID value to the IP address of the device, which is 177.162.4.24.

NOTE: **We strongly recommended that you configure the router identifier under the [edit routing-options] hierarchy level to avoid unpredictable behavior if the interface address on a loopback interface changes.**

Configuration

CLI Quick Configuration

To quickly configure an OSPF router identifier, copy the following command and paste it into the CLI.

```
[edit]
set routing-options router-id 177.162.4.24
```

Step-by-Step Procedure

To configure an OSPF router identifier:

1. Configure the OSPF router identifier by entering the **[router-id]** configuration value.

   ```
   [edit]
   user@host# set routing-options router-id 177.162.4.24
   ```

2. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Results Confirm your configuration by entering the **show routing-options router-id** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show routing-options router-id
router-id 177.162.4.24;
```

Verification

After you configure the router ID and activate OSPF on the routing device, the router ID is referenced by multiple OSPF operational mode commands that you can use to monitor and troubleshoot the OSPF protocol. The router ID fields are clearly marked in the output.

Related Documentation

- OSPF Configuration Overview on page 165
- OSPF Designated Router Overview on page 166
- Example: Controlling OSPF Designated Router Election on page 168

Example: Controlling OSPF Designated Router Election

This example shows how to control OSPF designated router election.

- Requirements on page 168
- Overview on page 168
- Configuration on page 168
- Verification on page 169

Requirements

Before you begin:

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the router identifiers for the devices in your OSPF network. See "Example: Configuring an OSPF Router Identifier" on page 167.

Overview

This example shows how to control OSPF designated router election. Within the example, you set the OSPF interface to **ge-/0/0/1** and the device priority to 200. The higher the priority value, the greater likelihood the routing device will become the designated router.

By default, routing devices have a priority of 128. A priority of 0 marks the routing device as ineligible to become the designated router. A priority of 1 means the routing device has the least chance of becoming a designated router.

Configuration

CLI Quick Configuration To quickly configure an OSPF designated router election, copy the following command and paste it into the CLI.

 Copyright © 2012, Juniper Networks, Inc.

```
[edit]
set protocols ospf area 0.0.0.3 interface ge-0/0/1 priority 200
```

Step-by-Step Procedure

To control OSPF designated router election:

1. Configure an OSPF interface and specify the device priority.

   NOTE: **To specify an OSPFv3 interface, include the ospf3 statement at the [edit protocols] hierarchy level.**

   ```
   [edit]
   user@host# set protocols ospf area 0.0.0.3 interface ge-0/0/1 priority 200
   ```

2. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Results

Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show protocols ospf
area 0.0.0.3 {
   interface ge-0/0/1.0 {
      priority 200;
   }
}
```

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

Verification

Confirm that the configuration is working properly.

- Verifying the Designated Router Election on page 169

Verifying the Designated Router Election

Purpose

Based on the priority you configured for a specific OSPF interface, you can confirm the address of the area's designated router. The DR ID, DR, or DR-ID field displays the address of the area's designated router. The BDR ID, BDR, or BDR-ID field displays the address of the backup designated router.

Action

From operational mode, enter the **show ospf interface** and the **show ospf neighbor** commands for OSPFv2, and enter the **show ospf3 interface** and the **show ospf3 neighbor** commands for OSPFv3.

Related Documentation

- OSPF Configuration Overview on page 165
- OSPF Designated Router Overview on page 166

- Example: Configuring an OSPF Router Identifier on page 167

OSPF Areas, Area Border Routers, and Backbone Areas

- Understanding OSPF Areas and Backbone Areas on page 170
- Example: Configuring a Single-Area OSPF Network on page 172
- Example: Configuring a Multiarea OSPF Network on page 174

Understanding OSPF Areas and Backbone Areas

OSPF networks in an autonomous system (AS) are administratively grouped into *areas*. Each area within an AS operates like an independent network and has a unique 32-bit area ID, which functions similar to a network address. Within an area, the topology database contains only information about the area, link-state advertisements (LSAs) are flooded only to nodes within the area, and routes are computed only within the area. The topology of an area is hidden from the rest of the AS, thus significantly reducing routing traffic in the AS. Subnetworks are divided into other areas, which are connected to form the whole of the main network. Routing devices that are wholly within an area are called *internal routers*. All interfaces on internal routers are directly connected to networks within the area.

The central area of an AS, called the *backbone area*, has a special function and is always assigned the area ID 0.0.0.0. (Within a simple, single-area network, this is also the ID of the area.) Area IDs are unique numeric identifiers, in dotted decimal notation, but they are not IP addresses. Area IDs need only be unique within an AS. All other networks or areas in the AS must be directly connected to the backbone area by a routing device that has interfaces in more than one area. These connecting routing devices are called *area border routers* (ABRs). Figure 27 on page 170 shows an OSPF topology of three areas connected by two ABRs.

Figure 27: Multiarea OSPF Topology

Copyright © 2012, Juniper Networks, Inc.

Because all areas are adjacent to the backbone area, OSPF routers send all traffic not destined for their own area through the backbone area. The ABRs in the backbone area are then responsible for transmitting the traffic through the appropriate ABR to the destination area. The ABRs summarize the link-state records of each area and advertise destination address summaries to neighboring areas. The advertisements contain the ID of the area in which each destination lies, so that packets are routed to the appropriate ABR. For example, in the OSPF areas shown in Figure 27 on page 170, packets sent from Router A to Router C are automatically routed through ABR B.

Junos OS supports active backbone detection. Active backbone detection is implemented to verify that ABRs are connected to the backbone. If the connection to the backbone area is lost, then the routing device's default metric is not advertised, effectively rerouting traffic through another ABR with a valid connection to the backbone. Active backbone detection enables transit through an ABR with no active backbone connection. An ABR advertises to other routing devices that it is an ABR even if the connection to the backbone is down, so that the neighbors can consider it for interarea routes.

An OSPF restriction requires all areas to be directly connected to the backbone area so that packets can be properly routed. All packets are routed first to the backbone area by default. Packets that are destined for an area other than the backbone area are then routed to the appropriate ABR and on to the remote host within the destination area.

In large networks with many areas, in which direct connectivity between all areas and the backbone area is physically difficult or impossible, you can configure virtual links to connect noncontiguous areas. Virtual links use a transit area that contains two or more ABRs to pass network traffic from one adjacent area to another. For example, Figure 28 on page 171 shows a virtual link between a noncontiguous area and the backbone area through an area connected to both.

Figure 28: OSPF Topology with a Virtual Link

In the topology shown in Figure 28 on page 171, a virtual link is established between area 0.0.0.3 and the backbone area through area 0.0.0.2. All outbound traffic destined for other areas is routed through area 0.0.0.2 to the backbone area and then to the appropriate ABR. All inbound traffic destined for area 0.0.0.3 is routed to the backbone area and then through area 0.0.0.2.

Related Documentation

- Example: Configuring a Single-Area OSPF Network on page 172
- Example: Configuring a Multiarea OSPF Network on page 174

Example: Configuring a Single-Area OSPF Network

This example shows how to configure a single-area OSPF network.

- Requirements on page 172
- Overview on page 172
- Configuration on page 173
- Verification on page 174

Requirements

Before you begin:

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the router identifiers for the devices in your OSPF network. See "Example: Configuring an OSPF Router Identifier" on page 167.

Overview

To activate OSPF on a network, you must enable the OSPF protocol on all interfaces within the network on which OSPF traffic is to travel. To enable OSPF, you must configure one or more interfaces on the device within an OSPF area. Once the interfaces are configured, OSPF LSAs are transmitted on all OSPF-enabled interfaces, and the network topology is shared throughout the network.

In an autonomous system (AS), the backbone area is always assigned area ID 0.0.0.0 (within a simple, single-area network, this is also the ID of the area). Area IDs are unique numeric identifiers, in dotted decimal notation. Area IDs need only be unique within an AS. All other networks or areas in the AS must be directly connected to the backbone area by area border routers that have interfaces in more than one area. You must also create a backbone area if your network consists of multiple areas. In this example, you create the backbone area and add interfaces, such as **ge-0/0/0**, as needed to the OSPF area.

To use OSPF on the device, you must configure at least one OSPF area, such as the one shown in Figure 29 on page 173.

Copyright © 2012, Juniper Networks, Inc.

Figure 29: Typical Single-Area OSPF Network Topology

Configuration

CLI Quick Configuration

To quickly configure a single-area OSPF network, copy the following command and paste it into the CLI. You repeat this configuration for all interfaces that are part of the OSPF area.

```
[edit]
set protocols ospf area 0.0.0.0 interface ge-0/0/0
```

Step-by-Step Procedure

To configure a single-area OSPF network:

1. Configure the single-area OSPF network by specifying the area ID and associated interface.

   NOTE: For a single-area OSPFv3 network, include the **ospf3** statement at the **[edit protocols]** hierarchy level.

   ```
   [edit]
   user@host# set protocols ospf area 0.0.0.0 interface ge-0/0/0
   ```

2. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Results

Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show protocols ospf
area 0.0.0.0 {
   interface ge-0/0/0.0;
}
```

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

Verification

Confirm that the configuration is working properly.

- Verifying the Interfaces in the Area on page 174

Verifying the Interfaces in the Area

Purpose Verify that the interface for OSPF or OSPFv3 has been configured for the appropriate area. Confirm that the Area field displays the value that you configured.

Action From operational mode, enter the **show ospf interface** command for OSPFv2, and enter the **show ospf3 interface** command for OSPFv3.

Related Documentation

- OSPF Configuration Overview on page 165
- Understanding OSPF Areas and Backbone Areas on page 170

Example: Configuring a Multiarea OSPF Network

This example shows how to configure a multiarea OSPF network. To reduce traffic and topology maintenance for the devices in an OSPF autonomous system (AS), you can group the OSPF-enabled routing devices into multiple areas.

- Requirements on page 174
- Overview on page 174
- Configuration on page 175
- Verification on page 176

Requirements

Before you begin:

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the router identifiers for the devices in your OSPF network. See "Example: Configuring an OSPF Router Identifier" on page 167.
- Control OSPF designated router election. See "Example: Controlling OSPF Designated Router Election" on page 168
- Configure a single-area OSPF network. See "Example: Configuring a Single-Area OSPF Network" on page 172.

Overview

To activate OSPF on a network, you must enable the OSPF protocol on all interfaces within the network on which OSPF traffic is to travel. To enable OSPF, you must configure one or more interfaces on the device within an OSPF area. Once the interfaces are configured, OSPF LSAs are transmitted on all OSPF-enabled interfaces, and the network topology is shared throughout the network.

 Copyright © 2012, Juniper Networks, Inc.

Each OSPF area consists of routing devices configured with the same area number. In Figure 30 on page 175, Router B resides in the backbone area of the AS. The backbone area is always assigned area ID 0.0.0.0. (All area IDs must be unique within an AS.) All other networks or areas in the AS must be directly connected to the backbone area by a router that has interfaces in more than one area. In this example, these area border routers are A, C, D, and E. You create an additional area (area 2) and assign it unique area ID 0.0.0.2, and then add interface **ge-0/0/0** to the OSPF area.

To reduce traffic and topology maintenance for the devices in an OSPF AS, you can group them into multiple areas as shown in Figure 30 on page 175.

Figure 30: Typical Multiarea OSPF Network Topology

Configuration

CLI Quick Configuration

To quickly configure a multiarea OSPF network, copy the following command and paste it into the CLI. You repeat this configuration for all interfaces that are part of the OSPF area.

```
[edit]
set protocols ospf area 0.0.0.2 interface ge-0/0/0
```

Step-by-Step Procedure

To configure a multiarea OSPF network:

1. Configure an additional area for your OSPF network.

   NOTE: **For a multiarea OSPFv3 network, include the ospf3 statement at the [edit protocols] hierarchy level.**

   ```
   [edit]
   user@host# set protocols ospf area 0.0.0.2 interface ge-0/0/0
   ```

2. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Copyright © 2012, Juniper Networks, Inc.

Results Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show protocols ospf
area 0.0.0.2 {
   interface ge-0/0/0.0;
}
```

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

Verification

Confirm that the configuration is working properly.

- Verifying the Interfaces in the Area on page 176

Verifying the Interfaces in the Area

Purpose Verify that the interface for OSPF or OSPFv3 has been configured for the appropriate area. Confirm that the Area field displays the value that you configured.

Action From operational mode, enter the **show ospf interface** command for OSPFv2, and enter the **show ospf3 interface** command for OSPFv3.

Related Documentation

- OSPF Configuration Overview on page 165
- Understanding OSPF Areas and Backbone Areas on page 170

OSPF Stub Areas and Not-So-Stubby Areas

- Understanding OSPF Stub Areas, Totally Stubby Areas, and Not-So-Stubby Areas on page 176
- Example: Configuring OSPF Stub and Totally Stubby Areas on page 178
- Example: Configuring OSPF Not-So-Stubby Areas on page 182

Understanding OSPF Stub Areas, Totally Stubby Areas, and Not-So-Stubby Areas

Figure 31 on page 177 shows an autonomous system (AS) across which many external routes are advertised. If external routes make up a significant portion of a topology database, you can suppress the advertisements in areas that do not have links outside the network. By doing so, you can reduce the amount of memory the nodes use to maintain the topology database and free it for other uses.

 Copyright © 2012, Juniper Networks, Inc.

Figure 31: OSPF AS Network with Stub Areas and NSSAs

To control the advertisement of external routes into an area, OSPF uses stub areas. By designating an area border router (ABR) interface to the area as a stub interface, you suppress external route advertisements through the ABR. Instead, the ABR advertises a default route (through itself) in place of the external routes and generates network summary (Type 3) link-state advertisements (LSAs). Packets destined for external routes are automatically sent to the ABR, which acts as a gateway for outbound traffic and routes the traffic appropriately.

NOTE: You must explicitly configure the ABR to generate a default route when attached to a stub or not-so-stubby-area (NSSA). To inject a default route with a specified metric value into the area, you must configure the **default-metric** option and specify a metric value.

For example, area 0.0.0.3 in Figure 31 on page 177 is not directly connected to the outside network. All outbound traffic is routed through the ABR to the backbone and then to the destination addresses. By designating area 0.0.0.3 as a stub area, you reduce the size of the topology database for that area by limiting the route entries to only those routes internal to the area.

A stub area that only allows routes internal to the area and restricts Type 3 LSAs from entering the stub area is often called a *totally stubby area*. You can convert area 0.0.0.3 to a totally stubby area by configuring the ABR to only advertise and allow the default route to enter into the area. External routes and destinations to other areas are no longer summarized or allowed into a totally stubby area.

NOTE: If you incorrectly configure a totally stubby area, you might encounter network connectivity issues. You should have advanced knowledge of OSPF and understand your network environment before configuring totally stubby areas.

Copyright © 2012, Juniper Networks, Inc.

Similar to area 0.0.0.3 in Figure 31 on page 177, area 0.0.0.4 has no external connections. However, area 0.0.0.4 has static customer routes that are not internal OSPF routes. You can limit the external route advertisements to the area and advertise the static customer routes by designating the area an NSSA. In an NSSA, the AS boundary router generates NSSA external (Type 7) LSAs and floods them into the NSSA, where they are contained. Type 7 LSAs allow an NSSA to support the presence of AS boundary routers and their corresponding external routing information. The ABR converts Type 7 LSAs into AS external (Type 5) LSAs and leaks them to the other areas, but external routes from other areas are not advertised within the NSSA.

Related Documentation

- OSPF Configuration Overview on page 165
- Example: Configuring OSPF Stub and Totally Stubby Areas on page 178
- Example: Configuring OSPF Not-So-Stubby Areas on page 182

Example: Configuring OSPF Stub and Totally Stubby Areas

This example shows how to configure an OSPF stub area and a totally stubby area to control the advertisement of external routes into an area.

- Requirements on page 178
- Overview on page 178
- Configuration on page 180
- Verification on page 181

Requirements

Before you begin:

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the router identifiers for the devices in your OSPF network. See "Example: Configuring an OSPF Router Identifier" on page 167.
- Control OSPF designated router election. See "Example: Controlling OSPF Designated Router Election" on page 168
- Configure a multiarea OSPF network. See "Example: Configuring a Multiarea OSPF Network" on page 174.

Overview

The backbone area, which is 0 in Figure 32 on page 180, has a special function and is always assigned the area ID 0.0.0.0. Area IDs are unique numeric identifiers, in dotted decimal notation. Area IDs need only be unique within an autonomous system (AS). All other networks or areas (such as 3, 7, and 9) in the AS must be directly connected to the backbone area by area border routers (ABRs) that have interfaces in more than one area.

Stub areas are areas through which or into which OSPF does not flood AS external link-state advertisements (Type 5 LSAs). You might create stub areas when much of

Copyright © 2012, Juniper Networks, Inc.

the topology database consists of AS external advertisements and you want to minimize the size of the topology databases on the internal routers in the stub area.

The following restrictions apply to stub areas:

- You cannot create a virtual link through a stub area.
- A stub area cannot contain an AS boundary router.
- You cannot configure the backbone as a stub area.
- You cannot configure an area as both a stub area and an not-so-stubby area (NSSA).

In this example, you configure each routing device in area 7 (area ID 0.0.0.7) as a stub router and some additional settings on the ABR:

- **stub**—Specifies that this area become a stub area and not be flooded with Type 5 LSAs. You must include the **stub** statement on all routing devices that are in area 7 because this area has no external connections.
- **default-metric**—Configures the ABR to generate a default route with a specified metric into the stub area. This default route enables packet forwarding from the stub area to external destinations. You configure this option only on the ABR. The ABR does not automatically generate a default route when attached to a stub. You must explicitly configure this option to generate a default route.
- **no-summaries**—(Optional) Prevents the ABR from advertising summary routes into the stub area by converting the stub area into a totally stubby area. If configured in combination with the **default-metric** statement, a totally stubby area only allows routes internal to the area and advertises the default route into the area. External routes and destinations to other areas are no longer summarized or allowed into a totally stubby area. Only the ABR requires this additional configuration because it is the only routing device within the totally stubby area that creates Type 3 LSAs used to receive and send traffic from outside of the area.

NOTE:

In Junos OS Release 8.5 and later, the following applies:

- A router-identifier interface that is not configured to run OSPF is no longer advertised as a stub network in OSPF LSAs.
- OSPF advertises a local route with a prefix length of 32 as a stub link if the loopback interface is configured with a prefix length other than 32. OSPF also advertises the direct route with the configured mask length, as in earlier releases.

Figure 32: OSPF Network Topology with Stub Areas and NSSAs

Configuration

CLI Quick Configuration

- To quickly configure an OSPF stub area, copy the following command and paste it into the CLI. You must configure all routing devices that are part of the stub area.

  ```
  [edit]
  set protocols ospf area 0.0.0.7 stub
  ```

- To quickly configure the ABR to inject a default route into the area, copy the following command and paste it into the CLI. You apply this configuration only on the ABR.

  ```
  [edit]
  set protocols ospf area 0.0.0.7 stub default-metric 10
  ```

- (Optional) To quickly configure the ABR to restrict all summary advertisements and allow only internal routes and default route advertisements into the area, copy the following command and paste it into the CLI. You apply this configuration only on the ABR.

  ```
  [edit]
  set protocols ospf area 0.0.0.7 stub no-summaries
  ```

Step-by-Step Procedure

To configure OSPF stub areas:

1. On all routing devices in the area, configure an OSPF stub area.

   NOTE: **To specify an OSPFv3 stub area, include the ospf3 statement at the [edit protocols] hierarchy level.**

   ```
   [edit]
   user@host# set protocols ospf area 0.0.0.7 stub
   ```

2. On the ABR, inject a default route into the area.

Copyright © 2012, Juniper Networks, Inc.

```
[edit]
user@host# set protocols ospf area 0.0.0.7 stub default-metric 10
```

3. (Optional) On the ABR, restrict summary LSAs from entering the area. This step converts the stub area into a totally stubby area.

```
[edit]
user@host# set protocols ospf area 0.0.0.7 stub no-summaries
```

4. If you are done configuring the devices, commit the configuration.

```
[edit]
user@host# commit
```

Results Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Configuration on all routing devices:

```
user@host# show protocols ospf
area 0.0.0.7 {
   stub;
}
```

Configuration on the ABR (the output also includes the optional setting):

```
user@host# show protocols ospf
area 0.0.0.7 {
   stub default-metric 10 no-summaries;
}
```

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

Verification

Confirm that the configuration is working properly.

- Verifying the Interfaces in the Area on page 181
- Verifying the Type of OSPF Area on page 181

Verifying the Interfaces in the Area

Purpose Verify that the interface for OSPF has been configured for the appropriate area. Confirm that the output includes Stub as the type of OSPF area.

Action From operational mode, enter the **show ospf interface detail** command for OSPFv2, and enter the **show ospf3 interface detail** command for OSPFv3.

Verifying the Type of OSPF Area

Purpose Verify that the OSPF area is a stub area. Confirm that the output displays Normal Stub as the Stub type.

Action From operational mode, enter the **show ospf overview** command for OSPFv2, and enter the **show ospf3 overview** command for OSPFv3.

Related Documentation

- OSPF Configuration Overview on page 165
- Understanding OSPF Stub Areas, Totally Stubby Areas, and Not-So-Stubby Areas on page 176
- Example: Configuring OSPF Not-So-Stubby Areas on page 182

Example: Configuring OSPF Not-So-Stubby Areas

This example shows how to configure an OSPF not-so-stubby area (NSSA) to control the advertisement of external routes into an area.

- Requirements on page 182
- Overview on page 182
- Configuration on page 184
- Verification on page 186

Requirements

Before you begin:

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the router identifiers for the devices in your OSPF network. See "Example: Configuring an OSPF Router Identifier" on page 167.
- Control OSPF designated router election. See "Example: Controlling OSPF Designated Router Election" on page 168
- Configure a multiarea OSPF network. See "Example: Configuring a Multiarea OSPF Network" on page 174.

Overview

The backbone area, which is 0 in Figure 33 on page 184, has a special function and is always assigned the area ID 0.0.0.0. Area IDs are unique numeric identifiers, in dotted decimal notation. Area IDs need only be unique within an AS. All other networks or areas (such as 3, 7, and 9) in the AS must be directly connected to the backbone area by ABRs that have interfaces in more than one area.

An OSPF stub area has no external routes, so you cannot redistribute routes from another protocol into a stub area. OSPF NSSAs allow external routes to be flooded within the area.

In addition, you might have a situation when exporting Type 7 LSAs into the NSSA is unnecessary. When an AS boundary router is also an ABR with an NSSA attached, Type 7 LSAs are exported into the NSSA by default. If the ABR is attached to multiple NSSAs, a separate Type 7 LSA is exported into each NSSA by default. During route redistribution,

 Copyright © 2012, Juniper Networks, Inc.

this routing device generates both Type 5 LSAs and Type 7 LSAs. You can disable exporting Type 7 LSAs into the NSSA.

NOTE: **The following restriction applies to NSSAs: You cannot configure an area as both a stub area and an NSSA.**

You configure each routing device in area 9 (area ID 0.0.0.9) with the following setting:

- **nssa**—Specifies an OSPF NSSA. You must include the **nssa** statement on all routing devices in area 9 because this area only has external connections to static routes.

You also configure the ABR in area 9 with the following additional settings:

- **no-summaries**—Prevents the ABR from advertising summary routes into the NSSA. If configured in combination with the **default-metric** statement, the NSSA only allows routes internal to the area and advertises the default route into the area. External routes and destinations to other areas are no longer summarized or allowed into the NSSA. Only the ABR requires this additional configuration because it is the only routing device within the NSSA that creates Type 3 LSAs used to receive and send traffic from outside the area.
- **default-lsa**—Configures the ABR to generate a default route into the NSSA. In this example, you configure the following:
  - **default-metric**—Specifies that the ABR generate a default route with a specified metric into the NSSA. This default route enables packet forwarding from the NSSA to external destinations. You configure this option only on the ABR. The ABR does not automatically generate a default route when attached to an NSSA. You must explicitly configure this option for the ABR to generate a default route.
  - **metric-type**—(Optional) Specifies the external metric type for the default LSA, which can be either Type 1 or Type 2. When OSPF exports route information from external ASs, it includes a cost, or external metric, in the route. The difference between the two metrics is how OSPF calculates the cost of the route. Type 1 external metrics are equivalent to the link-state metric, where the cost is equal to the sum of the internal costs plus the external cost. Type 2 external metrics use only the external cost assigned by the AS boundary router. By default, OSPF uses the Type 2 external metric.
  - **type-7**—(Optional) Floods Type 7 default LSAs into the NSSA if the **no-summaries** statement is configured. By default, when the **no-summaries** statement is configured, a Type 3 LSA is injected into NSSAs for Junos OS release 5.0 and later. To support backward compatibility with earlier Junos OS releases, include the **type-7** statement.

The second example also shows the optional configuration required to disable exporting Type 7 LSAs into the NSSA by including the **no-nssa-abr** statement on the routing device that performs the functions of both an ABR and an AS boundary router.

Copyright © 2012, Juniper Networks, Inc.

Figure 33: OSPF Network Topology with Stub Areas and NSSAs

Configuration

- Configuring Routing Devices to Participate in a Not-So-Stubby-Area on page 184
- Disabling the Export of Type 7 Link State Advertisements into Not-So-Stubby Areas on page 186

Configuring Routing Devices to Participate in a Not-So-Stubby-Area

CLI Quick Configuration

To quickly configure an OSPF NSSA, copy the following command and paste it into the CLI. You must configure all routing devices that are part of the NSSA.

```
[edit]
set protocols ospf area 0.0.0.9 nssa
```

To quickly configure an ABR that participates in an OSPF NSSA, copy the following commands and paste them into the CLI.

```
[edit]
set protocols ospf area 0.0.0.9 nssa default-lsa default-metric 10
set protocols ospf area 0.0.0.9 nssa default-lsa metric-type 1
set protocols ospf area 0.0.0.9 nssa default-lsa type-7
set protocols ospf area 0.0.0.9 nssa no-summaries
```

Step-by-Step Procedure

To configure OSPF NSSAs:

1. On all routing devices in the area, configure an OSPF NSSA.

NOTE: To specify an OSPFv3 NSSA area, include the **ospf3** statement at the **[edit protocols]** hierarchy level.

```
[edit]
user@host# set protocols ospf area 0.0.0.9 nssa
```


Copyright © 2012, Juniper Networks, Inc.

2. On the ABR, enter OSPF configuration mode and specify the NSSA area 0.0.0.9 that you already created.

```
[edit ]
user@host# edit protocols ospf area 0.0.0.9 nssa
```

3. On the ABR, inject a default route into the area.

```
[edit protocols ospf area 0.0.0.9 nssa]
user@host# set default-lsa default-metric 10
```

4. (Optional) On the ABR, specify the external metric type for the default route.

```
[edit protocols ospf area 0.0.0.9 nssa]
user@host# set default-lsa metric-type 1
```

5. (Optional) On the ABR, specify the flooding of Type 7 LSAs.

```
[edit protocols ospf area 0.0.0.9 nssa]
user@host# set default-lsa type-7
```

6. On the ABR, restrict summary LSAs from entering the area.

```
[edit protocols ospf area 0.0.0.9 nssa]
user@host# set no-summaries
```

7. If you are done configuring the devices, commit the configuration.

```
[edit protocols ospf area 0.0.0.9 nssa]
user@host# commit
```

Results Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Configuration on all routing devices in the area:

```
user@host# show protocols ospf
area 0.0.0.9 {
   nssa;
}
```

Configuration on the ABR. The output also includes the optional **metric-type** and **type-7** statements.

```
user@host# show protocols ospf
   area 0.0.0.9 {
   nssa {
      default-lsa {
         default-metric 10;
         metric-type 1;
         type-7;
      }
      no-summaries;
   }
}
```

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

Disabling the Export of Type 7 Link State Advertisements into Not-So-Stubby Areas

CLI Quick Configuration

To quickly disable exporting Type 7 LSAs into the NSSA, copy the following command and paste it into the CLI. You configure this setting on an AS boundary router that is also an ABR with an NSSA area attached.

```
[edit]
set protocols ospf no-nssa-abr
```

Step-by-Step Procedure

You can configure this setting if you have an AS boundary router that is also an ABR with an NSSA area attached.

1. Disable exporting Type 7 LSAs into the NSSA.

   NOTE: **To specify OSPFv3, include the ospf3 statement at the [edit protocols] hierarchy level.**

   ```
   [edit]
   user@host# set protocols ospf no-nssa-abr
   ```

2. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Results

Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show protocols ospf
no-nssa-abr;
```

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

Verification

Confirm that the configuration is working properly.

- Verifying the Interfaces in the Area on page 186
- Verifying the Type of OSPF Area on page 187
- Verifying the Type of LSAs on page 187

Verifying the Interfaces in the Area

Purpose

Verify that the interface for OSPF has been configured for the appropriate area. Confirm that the output includes Stub NSSA as the type of OSPF area.

Action

From operational mode, enter the **show ospf interface detail** command for OSPFv2, and enter the **show ospf3 interface detail** command for OSPFv3.

Copyright © 2012, Juniper Networks, Inc.

Verifying the Type of OSPF Area

Purpose Verify that the OSPF area is a stub area. Confirm that the output displays Not so Stubby Stub as the Stub type.

Action From operational mode, enter the **show ospf overview** command for OSPFv2, and enter the **show ospf3 overview** command for OSPFv3.

Verifying the Type of LSAs

Purpose Verify the type of LSAs that are in the area. If you disabled exporting Type 7 LSAs into an NSSA, confirm that the Type field does not include NSSA as a type of LSA.

Action From operational mode, enter the **show ospf overview** command for OSPFv2, and enter the **show ospf3 overview** command for OSPFv3.

Related Documentation

- OSPF Configuration Overview on page 165
- Understanding OSPF Stub Areas, Totally Stubby Areas, and Not-So-Stubby Areas on page 176
- Example: Configuring OSPF Stub and Totally Stubby Areas on page 178

OSPF Authentication

- Understanding OSPFv2 Authentication on page 187
- Example: Configuring Simple Authentication for OSPFv2 Exchanges on page 189
- Example: Configuring MD5 Authentication for OSPFv2 Exchanges on page 191
- Example: Configuring a Transition of MD5 Keys on an OSPFv2 Interface on page 193
- Example: Configuring IPsec Authentication for an OSPF Interface on page 196

Understanding OSPFv2 Authentication

All OSPFv2 protocol exchanges can be authenticated to guarantee that only trusted routing devices participate in the autonomous system's routing. By default, OSPFv2 authentication is disabled.

NOTE: **OSPFv3 does not have a built-in authentication method and relies on IP Security (IPsec) to provide this functionality.**

You can enable the following authentication types:

- Simple authentication—Authenticates by using a plain-text password that is included in the transmitted packet. The receiving routing device uses an authentication key (password) to verify the packet.

- MD5 authentication—Authenticates by using an encoded MD5 checksum that is included in the transmitted packet. The receiving routing device uses an authentication key (password) to verify the packet.

  You define an MD5 key for each interface. If MD5 is enabled on an interface, that interface accepts routing updates only if MD5 authentication succeeds. Otherwise, updates are rejected. The routing device only accepts OSPFv2 packets sent using the same key identifier (ID) that is defined for that interface.

- IPsec authentication (beginning with Junos OS Release 8.3)—Authenticates OSPFv2 interfaces, the remote endpoint of a sham link, and the OSPFv2 virtual link by using manual security associations (SAs) to ensure that a packet's contents are secure between the routing devices. You configure the actual IPsec authentication separately.

  NOTE: **You can configure IPsec authentication together with either MD5 or simple authentication.**

  The following restrictions apply to IPsec authentication for OSPFv2:

  - Dynamic Internet Key Exchange (IKE) SAs are not supported.
  - Only IPsec transport mode is supported. Tunnel mode is not supported.
  - Because only bidirectional manual SAs are supported, all OSPFv2 peers must be configured with the same IPsec SA. You configure a manual bidirectional SA at the **[edit security ipsec]** hierarchy level.
  - You must configure the same IPsec SA for all virtual links with the same remote endpoint address, for all neighbors on OSPF nonbroadcast multiaccess (NBMA) or point-to-multipoint links, and for every subnet that is part of a broadcast link.
  - OSPFv2 peer interfaces are not supported.

Because OSPF performs authentication at the area level, all routing devices within the area must have the same authentication and corresponding password (key) configured. For MD5 authentication to work, both the receiving and transmitting routing devices must have the same MD5 key. In addition, a simple password and MD5 key are mutually exclusive. You can configure only one simple password, but multiple MD5 keys.

As part of your security measures, you can change MD5 keys. You can do this by configuring multiple MD5 keys, each with a unique key ID, and setting the date and time to switch to the new key. Each unique MD5 key has a unique ID. The ID is used by the receiver of the OSPF packet to determine which key to use for authentication. The key ID, which is required for MD5 authentication, specifies the identifier associated with the MD5 key.

Related Documentation

- Example: Configuring Simple Authentication for OSPFv2 Exchanges on page 189
- Example: Configuring MD5 Authentication for OSPFv2 Exchanges on page 191
- Example: Configuring a Transition of MD5 Keys on an OSPFv2 Interface on page 193
- Example: Configuring IPsec Authentication for an OSPF Interface on page 196

Copyright © 2012, Juniper Networks, Inc.

- IPsec Overview in the *Junos OS System Basics Configuration Guide*

Example: Configuring Simple Authentication for OSPFv2 Exchanges

This example shows how to enable simple authentication for OSPFv2 exchanges.

- Requirements on page 189
- Overview on page 189
- Configuration on page 189
- Verification on page 190

Requirements

Before you begin:

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the router identifiers for the devices in your OSPF network. See "Example: Configuring an OSPF Router Identifier" on page 167.
- Control OSPF designated router election. See "Example: Controlling OSPF Designated Router Election" on page 168
- Configure a single-area OSPF network. See "Example: Configuring a Single-Area OSPF Network" on page 172.
- Configure a multiarea OSPF network. See "Example: Configuring a Multiarea OSPF Network" on page 174.

Overview

Simple authentication uses a plain-text password that is included in the transmitted packet. The receiving routing device uses an authentication key (password) to verify the packet. Plain-text passwords are not encrypted and might be subject to packet interception. This method is the least secure and should only be used if network security is not your goal.

You can configure only one simple authentication key (password) on the routing device. The simple key can be from 1 through 8 characters and can include ASCII strings. If you include spaces, enclose all characters in quotation marks (" ").

In this example, you specify OSPFv2 interface **so-0/1/0** in area 0.0.0.0, set the authentication type to simple-password, and define the key as PssWd4.

Configuration

CLI Quick Configuration

To quickly configure simple authentication, copy the following command, removing any line breaks, and then paste the command into the CLI. You must configure all routing devices within the area with the same authentication and corresponding password.

```
[edit]
set protocols ospf area 0.0.0.0 interface so-0/1/0 authentication simple-password PssWd4
```

Step-by-Step Procedure

To enable simple authentication for OSPFv2 exchanges:

1. Create an OSPF area.

   [edit]
   user@host# **edit protocols ospf area 0.0.0.0**

2. Specify the interface.

   [edit protocols ospf area 0.0.0.0]
   user@host# **edit interface so-0/1/0**

3. Set the authentication type and the password.

   [edit protocols ospf area 0.0.0.0 interface so-0/1/0.0]
   user@host# **set authentication simple-password PssWd4**

4. If you are done configuring the device, commit the configuration.

   [edit protocols ospf area 0.0.0.0 interface so-0/1/0.0]
   user@host# **commit**

NOTE: Repeat this entire configuration on all peer OSPFv2 routing devices in the area.

Results

Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

NOTE: After you configure the password, you do not see the password itself. The output displays the encrypted form of the password you configured.

```
user@host# show protocols ospf
  area 0.0.0.0 {
  interface so-0/1/0.0 {
    authentication {
      simple-password "$9$-3dY4ZUHm5FevX-db2g"; ## SECRET-DATA
    }
  }
}
```

Verification

Confirm that the configuration is working properly.

- Verifying the Configured Authentication Method on page 191

Copyright © 2012, Juniper Networks, Inc.

Verifying the Configured Authentication Method

Purpose Verify that the authentication method for sending and receiving OSPF protocol packets is configured. The Authentication Type field displays Password when configured for simple authentication.

Action From operational mode, enter the **show ospf interface** and the **show ospf overview** commands.

Related Documentation

- Understanding OSPFv2 Authentication on page 187
- Example: Configuring MD5 Authentication for OSPFv2 Exchanges on page 191
- Example: Configuring a Transition of MD5 Keys on an OSPFv2 Interface on page 193

Example: Configuring MD5 Authentication for OSPFv2 Exchanges

This example shows how to enable MD5 authentication for OSPFv2 exchanges.

- Requirements on page 191
- Overview on page 191
- Configuration on page 192
- Verification on page 193

Requirements

Before you begin:

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the router identifiers for the devices in your OSPF network. See "Example: Configuring an OSPF Router Identifier" on page 167.
- Control OSPF designated router election. See "Example: Controlling OSPF Designated Router Election" on page 168
- Configure a single-area OSPF network. See "Example: Configuring a Single-Area OSPF Network" on page 172.
- Configure a multiarea OSPF network. See "Example: Configuring a Multiarea OSPF Network" on page 174.

Overview

MD5 authentication uses an encoded MD5 checksum that is included in the transmitted packet. The receiving routing device uses an authentication key (password) to verify the packet.

You define an MD5 key for each interface. If MD5 is enabled on an interface, that interface accepts routing updates only if MD5 authentication succeeds. Otherwise, updates are

Copyright © 2012, Juniper Networks, Inc.

rejected. The routing device only accepts OSPFv2 packets sent using the same key identifier (ID) that is defined for that interface.

In this example, you create the backbone area (area 0.0.0.0), specify OSPFv2 interface **so-0/2/0**, set the authentication type to md5, and then define the authentication key ID as 5 and the password as PssWd8.

Configuration

CLI Quick Configuration

To quickly configure MD5 authentication, copy the following command and paste it into the CLI.

```
[edit]
set protocols ospf area 0.0.0.0 interface so-0/2/0 authentication md5 5 key PssWd8
```

Step-by-Step Procedure

To enable MD5 authentication for OSPFv2 exchanges:

1. Create an OSPF area.

   ```
   [edit]
   user@host# edit protocols ospf area 0.0.0.0
   ```

2. Specify the interface.

   ```
   [edit protocols ospf area 0.0.0.0]
   user@host# edit interface so-0/2/0
   ```

3. Configure MD5 authentication and set a key ID and an authentication password.

   ```
   [edit protocols ospf area 0.0.0.0 interface s0-0/2/0.0]
   user@host# set authentication md5 5 key PssWd8
   ```

4. If you are done configuring the device, commit the configuration.

   ```
   [edit protocols ospf area 0.0.0.0 interface s0-0/2/0.0]
   user@host# commit
   ```

NOTE: Repeat this entire configuration on all peer OSPFv2 routing devices.

Results

Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

NOTE: After you configure the password, you do not see the password itself. The output displays the encrypted form of the password you configured.

```
user@host# show protocols ospf
area 0.0.0.0 {
  interface so-0/2/0.0 {
    authentication {
```


Copyright © 2012, Juniper Networks, Inc.

```
            md5 5 key "$9$pXXhuIhreWx-wQF9puBEh"; ## SECRET-DATA
        }
    }
}
```

Verification

Confirm that the configuration is working properly.

Verifying the Configured Authentication Method

Purpose Verify that the authentication method for sending and receiving OSPF protocol packets is configured. When configured for MD5 authentication, the Authentication Type field displays MD5, the Active key ID field displays the unique number you entered that identifies the MD5 key, and the Start time field displays the date as Start time 1970 Jan 01 00:00:00 PST. Do not be alarmed by this start time. This is the default start time that the routing device displays if the MD5 key is effective immediately.

Action From operational mode, enter the **show ospf interface** and the **show ospf overview** commands.

Related Documentation

- Understanding OSPFv2 Authentication on page 187
- Example: Configuring Simple Authentication for OSPFv2 Exchanges on page 189
- Example: Configuring a Transition of MD5 Keys on an OSPFv2 Interface on page 193

Example: Configuring a Transition of MD5 Keys on an OSPFv2 Interface

This example shows how to configure a transition of MD5 keys on an OSPFv2 interface.

- Requirements on page 193
- Overview on page 194
- Configuration on page 194
- Verification on page 196

Requirements

Before you begin:

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the router identifiers for the devices in your OSPF network. See "Example: Configuring an OSPF Router Identifier" on page 167.
- Control OSPF designated router election. See "Example: Controlling OSPF Designated Router Election" on page 168

- Configure a single-area OSPF network. See "Example: Configuring a Single-Area OSPF Network" on page 172.
- Configure a multiarea OSPF network. See "Example: Configuring a Multiarea OSPF Network" on page 174.

Overview

MD5 authentication uses an encoded MD5 checksum that is included in the transmitted packet. For MD5 authentication to work, both the receiving and transmitting routing devices must have the same MD5 key.

You define an MD5 key for each interface. If MD5 is enabled on an interface, that interface accepts routing updates only if MD5 authentication succeeds. Otherwise, updates are rejected. The routing device only accepts OSPFv2 packets sent using the same key identifier (ID) that is defined for that interface.

For increased security, you can configure multiple MD5 keys, each with a unique key ID, and set the date and time to switch to a new key. The receiver of the OSPF packet uses the ID to determine which key to use for authentication.

In this example, you configure new keys to take effect at 12:01 AM on the first day of the next three months on OSPFv2 interface **fe-0/0/1** in the backbone area (area 0.0.0.0), and you configure the following MD5 authentication settings:

- **md5**—Specifies the MD5 authentication key ID. The key ID can be set to any value between 0 and 255, with a default value of 0. The routing device only accepts OSPFv2 packets sent using the same key ID that is defined for that interface.
- **key**—Specifies the MD5 key. Each key can be a value from 1 through 16 characters long. Characters can include ASCII strings. If you include spaces, enclose all characters in quotation marks (" ").
- **start-time**—Specifies the time to start using the MD5 key. This option enables you to configure a smooth transition mechanism for multiple keys. The start time is relevant for transmission but not for receiving OSPF packets.

NOTE: **You must set the same passwords and transition dates and times on all devices in the area so that OSPFv2 adjacencies remain active.**

Configuration

CLI Quick Configuration

To quickly configure multiple MD5 keys on an OSPFv2 interface, copy the following commands, remove any line breaks, and then paste the commands into the CLI.

```
[edit]
set protocols ospf area 0.0.0.0 interface fe-0/1/0 authentication md5 1 key $2010HaL
set protocols ospf area 0.0.0.0 interface fe-0/1/0 authentication md5 2 key NeWpsswdFEB
   start-time 2011-02-01.00:01
set protocols ospf area 0.0.0.0 interface fe-0/1/0 authentication md5 3 key NeWpsswdMAR
   start-time 2011-03-01.00:01
```


Copyright © 2012, Juniper Networks, Inc.

```
set protocols ospf area 0.0.0.0 interface fe-0/1/0 authentication md5 4 key NeWpsswdAPR
  start-time 2011-04-01.00:01
```

Step-by-Step Procedure

To configure multiple MD5 keys on an OSPFv2 interface:

1. Create an OSPF area.

   ```
   [edit]
   user@host# edit protocols ospf area 0.0.0.0
   ```

2. Specify the interface.

   ```
   [edit protocols ospf area 0.0.0.0]
   user@host# edit interface fe-0/1/0
   ```

3. Configure MD5 authentication and set an authentication password and key ID.

   ```
   [edit protocols ospf area 0.0.0.0 interface fe-0/1/0.0]
   user@host# set authentication md5 1 key $2010HaL
   ```

4. Configure a new key to take effect at 12:01 AM on the first day of February, March, and April.

   You configure a new authentication password and key ID for each month.

   a. For the month of February, enter the following:

      ```
      [edit protocols ospf area 0.0.0.0 interface fe-0/1/0.0]
      user@host# set authentication md5 2 key NeWpsswdFEB start-time
        2011-02-01.00:01
      ```

   b. For the month of March, enter the following:

      ```
      [edit protocols ospf area 0.0.0.0 interface fe-0/1/0.0]
      user@host# set authentication md5 3 key NeWpsswdMAR start-time
        2011-03-01.00:01
      ```

   c. For the month of April, enter the following:

      ```
      [edit protocols ospf area 0.0.0.0 interface fe-0/1/0.0]
      user@host# set authentication md5 4 key NeWpsswdAPR start-time
        2011-04-01.00:01
      ```

5. If you are done configuring the device, commit the configuration.

   ```
   [edit protocols ospf area 0.0.0.0 interface fe-0/1/0.0]
   user@host# commit
   ```

NOTE: Repeat this entire configuration on all peer OSPFv2 routing devices.

Results

Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

NOTE: After you configure the password, you do not see the password itself. The output displays the encrypted form of the password you configured.

```
user@host# show protocols ospf
area 0.0.0.0 {
  interface fe-0/1/0.0 {
    authentication {
      md5 1 key "$9$wzs24JGDjk.2gfTQ3CAp0B1hy"; ## SECRET-DATA
      md5 2 key "$9$Q9gz39t1IcML7EcwgJZq.RhSylMN-b4oZDi" start-time
        "2011-2-1.00:01:00 -0800"; ## SECRET-DATA
      md5 3 key "$9$zjo2nCpIRSWXNhSs4ZG.mEcyreW2gaZGjCt" start-time
        "2011-3-1.00:01:00 -0800"; ## SECRET-DATA
      md5 4 key "$9$fQn90OReML1Rds4oiHBIEhSevMLXNVqm" start-time
        "2011-4-1.00:01:00 -0700"; ## SECRET-DATA
    }
  }
}
```

Verification

Confirm that the configuration is working properly.

Verifying the Configured Authentication Method

Purpose Verify that the authentication method for sending and receiving OSPF protocol packets is configured. When configured for MD5 authentication with a transition of keys, the Auth type field displays MD5, the Active key ID field displays the unique number you entered that identifies the MD5 key, and the Start time field displays the time at which the routing device starts using an MD5 key to authenticate OSPF packets transmitted on the interface you configured.

Action From operational mode, enter the **show ospf interface** and the **show ospf overview** commands.

Related Documentation

- Understanding OSPFv2 Authentication on page 187
- Example: Configuring Simple Authentication for OSPFv2 Exchanges on page 189
- Example: Configuring MD5 Authentication for OSPFv2 Exchanges on page 191

Example: Configuring IPsec Authentication for an OSPF Interface

This example shows how to enable IP Security (IPsec) authentication for an OSPF interface.

- Requirements on page 197
- Overview on page 197
- Configuration on page 199
- Verification on page 201

 Copyright © 2012, Juniper Networks, Inc.

Requirements

Before you begin:

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the router identifiers for the devices in your OSPF network. See "Example: Configuring an OSPF Router Identifier" on page 167.
- Control OSPF designated router election. See "Example: Controlling OSPF Designated Router Election" on page 168
- Configure a single-area OSPF network. See "Example: Configuring a Single-Area OSPF Network" on page 172.
- Configure a multiarea OSPF network. See "Example: Configuring a Multiarea OSPF Network" on page 174.

Overview

You can use IPsec authentication for both OSPFv2 and OSPFv3. You configure the actual IPsec authentication separately and apply it to the applicable OSPF configuration.

OSPFv2

Beginning with Junos OS Release 8.3, you can use IPsec authentication to authenticate OSPFv2 interfaces, the remote endpoint of a sham link, and the OSPFv2 virtual link by using manual security associations (SAs) to ensure that a packet's contents are secure between the routing devices.

NOTE: **You can configure IPsec authentication together with either MD5 or simple authentication.**

To enable IPsec authentication, do one of the following:

- For an OSPFv2 interface, include the **ipsec-sa *name*** statement for a specific interface:

  interface *interface-name* ipsec-sa *name*;

- For a remote sham link, include the **ispec-sa *name*** statement for the remote end point of the sham link:

  sham-link-remote *address* ipsec-sa *name*;

NOTE: **If a Layer 3 VPN configuration has multiple sham links with the same remote endpoint IP address, you must configure the same IPsec security association for all the remote endpoints. You configure a Layer 3 VPN at the [edit routing-instances *routing-instance-name* instance-type] hierarchy level. For more information about Layer 3 VPNs, see the *Junos OS VPNs Configuration Guide*.**

- For a virtual link, include the **ipsec-sa** ***name*** statement for a specific virtual link:

  virtual-link neighbor-id *router-id* transit-area *area-id* ipsec-sa *name*;

OSPFv3

OSPFv3 does not have a built-in authentication method and relies on IPsec to provide this functionality. You use IPsec authentication to secure OSPFv3 interfaces and protect OSPFv3 virtual links by using manual SAs to ensure that a packet's contents are secure between the routing devices.

To apply authentication, do one of the following:

- For an OSPFv3 interface, include the **ipsec-sa** ***name*** statement for a specific interface:

  interface *interface-name* ipsec-sa *name*;

- For a virtual link, include the **ipsec-sa** ***name*** statement for a specific virtual link:

  virtual-link neighbor-id *router-id* transit-area *area-id* ipsec-sa *name*;

Tasks to Complete for Both OSPFv2 and OSPFv3

In this example, you perform the following tasks:

1. Configure IPsec authentication. To do this, define a manual SA named **sa1** and specify the processing direction, the protocol used to protect IP traffic, the security parameter index (SPI), and the authentication algorithm and key.

   a. Configure the following option at the **[edit security ipsec security-association** ***sa-name*** **mode]** hierarchy level:

      transport—Specifies transport mode. This mode protects traffic when the communication endpoint and the cryptographic endpoint are the same. The data portion of the IP packet is encrypted, but the IP header is not.

   b. Configure the following option at the **[edit security ipsec security-association** ***sa-name*** **manual direction]** hierarchy level:

      bidirectional—Defines the direction of IPsec processing. By specifying bidrectional, the same algorithms, keys, and security paramater index (SPI) values you configure are used in both directions.

   c. Configure the following options at the **[edit security ipsec security-association** ***sa-name*** **manual direction bidirectional]** hierarchy level:

      protocol—Defines the IPsec protocol used by the manual SA to protect IP traffic. You can specify either the authentication header (AH) or the Encapsulating Security Payload (ESP). If you specify AH, which you do in this example, you cannot configure encryption.

      spi—Configures the SPI for the manual SA. An SPI is an arbitrary value that uniquely identifies which SA to use at the receiving host. The sending host uses the SPI to identify and select which SA to use to secure every packet. The receiving host uses the SPI to identify and select the encryption algorithm and key used to decrypt packets. In this example, you specify 256.

 Copyright © 2012, Juniper Networks, Inc.

authentication—Configures the authentication algorithm and key. The **algorithm** option specifies the hash algorithm that authenticates packet data. In this example, you specify **hmac-md5-96**, which produces a 128-bit digest. The **key** option indicates the type of authentication key. In this example, you specify **ascii-text-key**, which is 16 ASCII characters for the **hmac-md5-96** algorithm.

2. Enable IPsec authentication on OSPF interface **so-0/2/0.0** in the backbone area (area 0.0.0.0) by including the name of the manual SA **sa1** that you configured at the **[edit security ipsec]** hierarchy level.

Configuration

- Configuring Security Associations on page 199
- Enabling IPsec Authentication for an OSPF Interface on page 200

Configuring Security Associations

CLI Quick Configuration

To quickly configure a manual SA to be used for IPsec authentication on an OSPF interface, copy the following commands, remove any line breaks, and then paste the commands into the CLI.

```
[edit]
set security ipsec security-association sa1
set security ipsec security-association sa1 mode transport
set security ipsec security-association sa1 manual direction bidirectional
set security ipsec security-association sa1 manual direction bidirectional protocol ah
set security ipsec security-association sa1 manual direction bidirectional spi 256
set security ipsec security-association sa1 manual direction bidirectional authentication
  algorithm hmac-md5-96 key ascii-text 123456789012abc
```

Step-by-Step Procedure

To configure a manual SA to be used on an OSPF interface:

1. Specify a name for the SA.

   ```
   [edit]
   user@host# edit security ipsec security-association sa1
   ```

2. Specify the mode of the SA.

   ```
   [edit security ipsec security-association sa1 ]
   user@host# set mode transport
   ```

3. Configure the direction of the manual SA.

   ```
   [edit security ipsec security-association sa1 ]
   user@host# set manual direction bidirectional
   ```

4. Configure the IPsec protocol to use.

   ```
   [edit security ipsec security-association sa1 ]
   user@host# set manual direction bidirectional protocol ah
   ```

5. Configure the value of the SPI.

   ```
   [edit security ipsec security-association sa1 ]
   user@host# set manual direction bidirectional spi 256
   ```

6. Configure the authentication algorithm and key.

```
[edit security ipsec security-association sa1 ]
user@host# set manual direction bidirectional authentication algorithm
  hmac-md5-96 key ascii-text 123456789012abc
```

7. If you are done configuring the device, commit the configuration.

```
[edit security ipsec security-association sa1 ]
user@host# commit
```

NOTE: Repeat this entire configuration on all peer OSPF routing devices.

Results Confirm your configuration by entering the **show security ipsec** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

NOTE: After you configure the password, you do not see the password itself. The output displays the encrypted form of the password you configured.

```
user@host# show security ipsec
security-association sa1 {
  mode transport;
  manual {
    direction bidirectional {
      protocol ah;
      spi 256;
      authentication {
        algorithm hmac-md5-96;
        key ascii-text "$9$AP5Hp1RcylMLxSygoZUHk1REhKMVwY2oJx7jHq.zF69A0OR";
          ## SECRET-DATA
      }
    }
  }
}
```

Enabling IPsec Authentication for an OSPF Interface

CLI Quick Configuration To quickly apply a manual SA used for IPsec authentication to an OSPF interface, copy the following command and paste it into the CLI.

```
[edit]
set protocols ospf area 0.0.0.0 interface so-0/2/0 ipsec-sa sa1
```

Step-by-Step Procedure To enable IPsec authentication for an OSPF interface:

1. Create an OSPF area.

NOTE: To specify OSPFv3, include the **ospf3** statement at the **[edit protocols]** hierarchy level.

 Copyright © 2012, Juniper Networks, Inc.

```
[edit]
user@host# edit protocols ospf area 0.0.0.0
```

2. Specify the interface.

```
[edit protocols ospf area 0.0.0.0]
user@host# edit interface so-0/2/0
```

3. Apply the IPsec manual SA.

```
[edit protocols ospf area 0.0.0.0 interface so-0/2/0.0]
user@host# set ipsec-sa sa1
```

4. If you are done configuring the device, commit the configuration.

```
[edit protocols ospf area 0.0.0.0 interface so-0/2/0.0]
user@host# commit
```

NOTE: **Repeat this entire configuration on all peer OSPF routing devices.**

Results Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show protocols ospf
area 0.0.0.0 {
   interface so-0/2/0.0 {
      ipsec-sa sa1;
   }
}
```

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

Verification

Confirm that the configuration is working properly.

- Verifying the IPsec Security Association Settings on page 201
- Verifying the IPsec Security Association on the OSPF Interface on page 202

Verifying the IPsec Security Association Settings

Purpose Verify the configured IPsec security association settings. Verify the following information:

- The Security association field displays the name of the configured security association.
- The SPI field displays the value you configured.
- The Mode field displays transport mode.
- The Type field displays manual as the type of security association.

Action From operational mode, enter the **show ipsec security-associations** command.

Verifying the IPsec Security Association on the OSPF Interface

Purpose Verify that the IPsec security association that you configured has been applied to the OSPF interface. Confirm that the IPSec SA name field displays the name of the configured IPsec security association.

Action From operational mode, enter the **show ospf interface detail** command for OSPFv2, and enter the **show ospf3 interface detail** command for OSPFv3.

Related Documentation

- Understanding OSPFv2 Authentication on page 187
- Understanding OSPFv3 Authentication
- Security Services Configuration Guidelines in the *Junos OS System Basics Configuration Guide*
- IPsec Services Configuration Guidelines in the *Junos OS Services Interfaces Configuration Guide*

OSPF Traffic Control

- Understanding OSPF Traffic Control on page 202
- Example: Controlling the Cost of Individual OSPF Network Segments on page 204
- Example: Controlling OSPF Route Preferences on page 208

Understanding OSPF Traffic Control

Once a topology is shared across the network, OSPF uses the topology to route packets between network nodes. Each path between neighbors is assigned a cost based on the throughput, round-trip time, and reliability of the link. The sum of the costs across a particular path between hosts determines the overall cost of the path. Packets are then routed along the shortest path using the shortest-path-first (SPF) algorithm. If multiple equal-cost paths exist between a source and destination address, OSPF routes packets along each path alternately, in round-robin fashion. Routes with lower total path metrics are preferred over those with higher path metrics.

You can use the following methods to control OSPF traffic:

- Control the cost of individual OSPF network segments
- Dynamically adjust OSPF interface metrics based on bandwidth
- Control OSPF route selection

Controlling the Cost of Individual OSPF Network Segments

OSPF uses the following formula to determine the cost of a route:

```
cost = reference-bandwidth / interface bandwidth
```

 Copyright © 2012, Juniper Networks, Inc.

You can modify the reference-bandwidth value, which is used to calculate the default interface cost. The interface bandwidth value is not user-configurable and refers to the actual bandwidth of the physical interface.

By default, OSPF assigns a default cost metric of 1 to any link faster than 100 Mbps, and a default cost metric of 0 to the loopback interface (**lo0**). No bandwidth is associated with the loopback interface.

To control the flow of packets across the network, OSPF allows you to manually assign a cost (or metric) to a particular path segment. When you specify a metric for a specific OSPF interface, that value is used to determine the cost of routes advertised from that interface. For example, if all routers in the OSPF network use default metric values, and you increase the metric on one interface to 5, all paths through that interface have a calculated metric higher than the default and are not preferred.

NOTE: **Any value you configure for the metric overrides the default behavior of using the reference-bandwidth value to calculate the route cost for that interface.**

Dynamically Adjusting OSPF Interface Metrics Based on Bandwidth

You can specify a set of bandwidth threshold values and associated metric values for an OSPF interface or for a topology on an OSPF interface. When the bandwidth of an interface changes, the Junos OS automatically sets the interface metric to the value associated with the appropriate bandwidth threshold value. Junos OS uses the smallest configured bandwidth threshold value that is equal to or greater than the actual interface bandwidth to determine the metric value. If the interface bandwidth is greater than any of the configured bandwidth threshold values, the metric value configured for the interface is used instead of any of the bandwidth-based metric values configured. The ability to recalculate the metric for an interface when its bandwidth changes is especially useful for aggregate interfaces.

NOTE: **You must also configure a metric for the interface when you enable bandwidth-based metrics.**

Controlling OSPF Route Preferences

You can control the flow of packets through the network using route preferences. Route preferences are used to select which route is installed in the forwarding table when several protocols calculate routes to the same destination. The route with the lowest preference value is selected.

By default, internal OSPF routes have a preference value of 10, and external OSPF routes have a preference value of 150. Although the default settings are appropriate for most environments, you might want to modify the default settings if all of the routing devices in your OSPF network use the default preference values, or if you are planning to migrate from OSPF to a different interior gateway protocol (IGP). If all of the devices use the

Copyright © 2012, Juniper Networks, Inc.

default route preference values, you can change the route preferences to ensure that the path through a particular device is selected for the forwarding table any time multiple equal-cost paths to a destination exist. When migrating from OSPF to a different IGP, modifying the route preferences allows you to perform the migration in a controlled manner.

Related Documentation

- OSPF Overview on page 160
- Example: Controlling the Cost of Individual OSPF Network Segments on page 204
- Example: Controlling OSPF Route Preferences on page 208
- *Junos OS Routing Protocols Configuration Guide*

Example: Controlling the Cost of Individual OSPF Network Segments

This example shows how to control the cost of individual OSPF network segments.

- Requirements on page 204
- Overview on page 204
- Configuration on page 205
- Verification on page 207

Requirements

Before you begin:

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the router identifiers for the devices in your OSPF network. See "Example: Configuring an OSPF Router Identifier" on page 167.
- Control OSPF designated router election. See "Example: Controlling OSPF Designated Router Election" on page 168
- Configure a single-area OSPF network. See "Example: Configuring a Single-Area OSPF Network" on page 172.

Overview

All OSPF interfaces have a cost, which is a routing metric that is used in the link-state calculation. Routes with lower total path metrics are preferred to those with higher path metrics. In this example, we explore how to control the cost of OSPF network segments.

By default, OSPF assigns a default cost metric of 1 to any link faster than 100 Mbps, and a default cost metric of 0 to the loopback interface (**lo0**). No bandwidth is associated with the loopback interface. This means that all interfaces faster than 100 Mbps have the same default cost metric of 1. If multiple equal-cost paths exist between a source and destination address, OSPF routes packets along each path alternately, in round-robin fashion.

Copyright © 2012, Juniper Networks, Inc.

Having the same default metric might not be a problem if all of the interfaces are running at the same speed. If the interfaces operate at different speeds, you might notice that traffic is not routed over the fastest interface because OSPF equally routes packets across the different interfaces. For example, if your routing device has Fast Ethernet and Gigabit Ethernet interfaces running OSPF, each of these interfaces have a default cost metric of 1.

In the first example, you set the reference bandwidth to 10g (10 Gbps, as denoted by 10,000,000,000 bits) by including the **reference-bandwidth** statement. With this configuration, OSPF assigns the Fast Ethernet interface a default metric of 100, and the Gigabit Ethernet interface a metric of 10. Since the Gigabit Ethernet interface has the lowest metric, OSPF selects it when routing packets. The range is 9600 through 1,000,000,000,000 bits.

Figure 34 on page 205 shows three routing devices in area 0.0.0.0 and assumes that the link between Device R2 and Device R3 is congested with other traffic. You can also control the flow of packets across the network by manually assigning a metric to a particular path segment. Any value you configure for the metric overrides the default behavior of using the reference-bandwidth value to calculate the route cost for that interface. To prevent the traffic from Device R3 going directly to Device R2, you adjust the metric on the interface on Device R3 that connects with Device R1 so that all traffic goes through Device R1.

In the second example, you set the metric to 5 on interface **fe-1/0/1** on Device R3 that connects with Device R1 by including the **metric** statement. The range is 1 through 65,535.

Figure 34: OSPF Metric Configuration

Configuration

- Configuring the Reference Bandwidth on page 205
- Configuring a Metric for a Specific OSPF Interface on page 206

Configuring the Reference Bandwidth

CLI Quick Configuration

To quickly configure the reference bandwidth, copy the following command and paste it into the CLI.

```
[edit]
set protocols ospf reference-bandwidth 10g
```

Step-by-Step Procedure

To configure the reference bandwidth:

1. Configure the reference bandwidth to calculate the default interface cost.

   NOTE: To specify OSPFv3, include the **ospf3** statement at the **[edit protocols]** hierarchy level.

   ```
   [edit]
   user@host# set protocols ospf reference-bandwidth 10g
   ```

   TIP: As a shortcut in this example, you enter **10g** to specify 10 Gbps reference bandwidth. Whether you enter **10g** or **10000000000**, the output of **show protocols ospf** command displays 10 Gbps as **10g**, not **10000000000**.

2. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

   NOTE: Repeat this entire configuration on all routing devices in a shared network.

Results

Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show protocols ospf
reference-bandwidth 10g;
```

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

Configuring a Metric for a Specific OSPF Interface

CLI Quick Configuration

To quickly configure a metric for a specific OSPF interface, copy the following command and paste it into the CLI.

```
[edit]
set protocols ospf area 0.0.0.0 interface fe-1/0/1 metric 5
```

Step-by-Step Procedure

To configure the metric for a specific OSPF interface:

1. Create an OSPF area.

Copyright © 2012, Juniper Networks, Inc.

NOTE: To specify OSPFv3, include the **ospf3** statement at the **[edit protocols]** hierarchy level.

```
[edit]
user@host# edit protocols ospf area 0.0.0.0
```

2. Configure the metric of the OSPF network segment.

```
[edit protocols ospf area 0.0.0.0 ]
user@host# set interface fe-1/0/1 metric 5
```

3. If you are done configuring the device, commit the configuration.

```
[edit protocols ospf area 0.0.0.0 ]
user@host# commit
```

Results Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show protocols ospf
area 0.0.0.0 {
   interface fe-1/0/1.0 {
      metric 5;
   }
}
```

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

Verification

Confirm that the configuration is working properly.

- Verifying the Configured Metric on page 207
- Verifying the Route on page 207

Verifying the Configured Metric

Purpose Verify the metric setting on the interface. Confirm that the Cost field displays the interface's configured metric (cost). When choosing paths to a destination, OSPF uses the path with the lowest cost.

Action From operational mode, enter the **show ospf interface detail** command for OSPFv2, and enter the **show ospf3 interface detail** command for OSPFv3.

Verifying the Route

Purpose When choosing paths to a destination, OSPF uses the path with the lowest total cost. Confirm that OSPF is using the appropriate path.

Action From operational mode, enter the **show route** command.

Related Documentation

- Understanding OSPF Traffic Control on page 202
- Example: Controlling OSPF Route Preferences on page 208

Example: Controlling OSPF Route Preferences

This example shows how to control OSPF route selection in the forwarding table. This example also shows how you might control route selection if you are migrating from OSPF to another IGP.

- Requirements on page 208
- Overview on page 208
- Configuration on page 209
- Verification on page 209

Requirements

This example assumes that OSPF is properly configured and running in your network, and you want to control route selection because you are planning to migrate from OSPF to a different IGP.

- Configure the device interfaces. See the *Junos OS Interfaces Fundamentals Configuration Guide* or the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Configure the IGP that you want to migrate to. See the *Junos OS Routing Protocols Configuration Guide*.

Overview

Route preferences are used to select which route is installed in the forwarding table when several protocols calculate routes to the same destination. The route with the lowest preference value is selected.

By default, internal OSPF routes have a preference value of 10, and external OSPF routes have a preference value of 150. You might want to modify this setting if you are planning to migrate from OSPF to a different IGP. Modifying the route preferences enables you to perform the migration in a controlled manner.

This example makes the following assumptions:

- OSPF is already running in your network.
- You want to migrate from OSPF to IS-IS.
- You configured IS-IS per your network requirements and confirmed it is working properly.

Copyright © 2012, Juniper Networks, Inc.

In this example, you increase the OSPF route preference values to make them less preferred than IS-IS routes by specifying 168 for internal OSPF routes and 169 for external OSPF routes. IS-IS internal routes have a preference of either 15 (for Level1) or 18 (for Level 2), and external routes have a preference of 160 (for Level 1) or 165 (for Level 2). In general, it is preferred to leave the new protocol at its default settings to minimize complexities and simplify any future addition of routing devices to the network. To modify the OSPF route preference values, configure the following settings:

- **preference**—Specifies the route preference for internal OSPF routes. By default, internal OSPF routes have a value of 10. The range is from 0 through 4,294967,295 ($2^{32} - 1$).
- **external-preference**—Specifies the route preference for external OSPF routes. By default, external OSPF routes have a value of 150. The range is from 0 through 4,294967,295 ($2^{32} - 1$).

Configuration

CLI Quick Configuration

To quickly configure the OSPF route preference values, copy the following command and paste it into the CLI.

```
[edit]
set protocols ospf preference 168 external-preference 169
```

To configure route selection:

1. Enter OSPF configuration mode and set the external and internal routing preferences.

   NOTE: **To specify OSPFv3, include the ospf3 statement at the [edit protocols] hierarchy level.**

   ```
   [edit]
   user@host# set protocols ospf preference 168 external-preference 169
   ```

2. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Results

Confirm your configuration by entering the **show protocols ospf** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show protocols ospf
preference 168;
external-preference 169;
```

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

Verification

Confirm that the configuration is working properly.

- Verifying the Route on page 210

Verifying the Route

Purpose Verify that the IGP is using the appropriate route. After the new IGP becomes the preferred protocol (in this example, IS-IS), you should monitor the network for any issues. After you confirm that the new IGP is working properly, you can remove the OSPF configuration from the routing device by entering the **delete ospf** command at the **[edit protocols]** hierarchy level.

Action From operational mode, enter the **show route** command.

Related Documentation

- Understanding OSPF Traffic Control on page 202
- Example: Controlling the Cost of Individual OSPF Network Segments on page 204

Verifying an OSPF Configuration

To verify an OSPF configuration, perform these tasks:

- Verifying OSPF-Enabled Interfaces on page 210
- Verifying OSPF Neighbors on page 211
- Verifying the Number of OSPF Routes on page 211
- Verifying Reachability of All Hosts in an OSPF Network on page 213

Verifying OSPF-Enabled Interfaces

Purpose Verify that OSPF is running on a particular interface and that the interface is in the desired area.

Action From the CLI, enter the **show ospf interface** command.

Sample Output

```
user@host> show ospf interface
Intf                State     Area            DR ID           BDR ID        Nbrs
at-5/1/0.0          PtToPt   0.0.0.0         0.0.0.0         0.0.0.0         1
ge-2/3/0.0          DR       0.0.0.0         192.168.4.16    192.168.4.15    1
lo0.0               DR       0.0.0.0         192.168.4.16    0.0.0.0         0
so-0/0/0.0          Down     0.0.0.0         0.0.0.0         0.0.0.0         0
so-6/0/1.0          PtToPt   0.0.0.0         0.0.0.0         0.0.0.0         1
so-6/0/2.0          Down     0.0.0.0         0.0.0.0         0.0.0.0         0
so-6/0/3.0          PtToPt   0.0.0.0         0.0.0.0         0.0.0.0         1
```

Meaning The output shows a list of the device interfaces that are configured for OSPF. Verify the following information:

- Each interface on which OSPF is enabled is listed.
- Under **Area**, each interface shows the area for which it was configured.
- Under **Intf** and **State**, the device loopback (**lo0.0**) interface and LAN interface that are linked to the OSPF network's designated router (DR) are identified.

 Copyright © 2012, Juniper Networks, Inc.

- Under **DR ID**, the IP address of the OSPF network's designated router appears.
- Under **State**, each interface shows a state of **PtToPt** to indicate a point-to-point connection. If the state is **Waiting**, check the output again after several seconds. A state of **Down** indicates a problem.
- The designated router addresses always show a state of **DR**.

Verifying OSPF Neighbors

Purpose OSPF neighbors are interfaces that have an immediate adjacency. On a point-to-point connection between the device and another router running OSPF, verify that each router has a single OSPF neighbor.

Action From the CLI, enter the **show ospf neighbor** command.

Sample Output

```
user@host> show ospf neighbor
 Address          Intf                State      ID               Pri  Dead
192.168.254.225  fxp3.0               2Way      10.250.240.32    128   36
192.168.254.230  fxp3.0               Full      10.250.240.8     128   38
192.168.254.229  fxp3.0               Full      10.250.240.35    128   33
10.1.1.129       fxp2.0               Full      10.250.240.12    128   37
10.1.1.131       fxp2.0               Full      10.250.240.11    128   38
10.1.2.1         fxp1.0               Full      10.250.240.9     128   32
10.1.2.81        fxp0.0               Full      10.250.240.10    128   33
```

Meaning The output shows a list of the device's OSPF neighbors and their addresses, interfaces, states, router IDs, priorities, and number of seconds allowed for inactivity (“dead” time). Verify the following information:

- Each interface that is immediately adjacent to the device is listed.
- The device's own loopback address and the loopback addresses of any routers with which the device has an immediate adjacency are listed.
- Under **State**, each neighbor shows a state of **Full**. Because full OSPF connectivity is established over a series of packet exchanges between clients, the OSPF link might take several seconds to establish. During that time, the state might be displayed as **Attempt**, **Init**, or **2way**, depending on the stage of negotiation.

  If, after 30 seconds, the state is not **Full**, the OSPF configuration between the neighbors is not functioning correctly.

Verifying the Number of OSPF Routes

Purpose Verify that the OSPF routing table has entries for the following:

- Each subnetwork reachable through an OSPF link
- Each loopback address reachable on the network

For example, Figure 35 on page 212 shows a sample network with an OSPF topology.

Figure 35: Sample OSPF Network Topology

In this topology, OSPF is being run on all interfaces. Each segment in the network is identified by an address with a **/24** prefix, with interfaces on either end of the segment being identified by unique IP addresses.

Action From the CLI, enter the **show ospf route** command.

Sample Output

```
user@host> show ospf route
Prefix              Path   Route      NH   Metric  NextHop      Nexthop
                    Type   Type       Type        Interface    addr/label
10.10.10.1/24       Intra  Network    IP   1       ge-0/0/2.0   10.0.21.1
10.10.10.2/24       Intra  Network    IP   1       ge-0/0/2.0   10.0.21.1
10.10.10.4/24       Intra  Network    IP   1       ge-0/0/1.0   10.0.13.1
10.10.10.5/24       Intra  Network    IP   1       ge-0/0/2.0   10.0.21.1
10.10.10.6/24       Intra  Network    IP   1       ge-0/0/1.0   10.0.13.1
10.10.10.10/24      Intra  Network    IP   1       ge-0/0/2.0   10.0.21.1
10.10.10.11/24      Intra  Network    IP   1       ge-0/0/1.0   10.0.13.1
10.10.10.13/24      Intra  Network    IP   1       ge-0/0/1.0
10.10.10.16/24      Intra  Network    IP   1       ge-0/0/1.0   10.0.13.1
10.10.10.19/24      Intra  Network    IP   1       ge-0/0/1.0   10.0.13.1
10.10.10.20/24      Intra  Network    IP   1       ge-0/0/2.0   10.0.21.1
10.10.10.21/24      Intra  Network    IP   1       ge-0/0/2.0
192.168.5.1         Intra  Router     IP   1       ge-0/0/2.0   10.0.21.1
192.168.5.2         Intra  Router     IP   1       lo0
192.168.5.3         Intra  Router     IP   1       ge-0/0/1.0   10.0.13.1
192.168.5.4         Intra  Router     IP   1       ge-0/0/1.0   10.0.13.1
192.168.5.5         Intra  Router     IP   1       ge-0/0/1.0   10.0.13.1
192.168.5.6         Intra  Router     IP   1       ge-0/0/2.0   10.0.21.1
192.168.5.7         Intra  Router     IP   1       ge-0/0/2.0   10.0.21.1
192.168.5.8         Intra  Router     IP   1       ge-0/0/2.0   10.0.21.1
192.168.5.9         Intra  Router     IP   1       ge-0/0/1.0   10.0.13.1
```

Meaning The output lists each route, sorted by IP address. Routes are shown with a route type of **Network**, and loopback addresses are shown with a route type of **Router**.

For the example shown in Figure 35 on page 212, verify that the OSPF routing table has 21 entries, one for each network segment and one for each router's loopback address.

 Copyright © 2012, Juniper Networks, Inc.

Verifying Reachability of All Hosts in an OSPF Network

Purpose By using the traceroute tool on each loopback address in the network, verify that all hosts in the network are reachable from each device.

Action For each device in the OSPF network:

1. In the J-Web interface, select **Troubleshoot>Traceroute**.
2. In the Host Name box, type the name of a host for which you want to verify reachability from the device.
3. Click **Start**. Output appears on a separate page.

Sample Output

```
1 172.17.40.254 (172.17.40.254) 0.362 ms 0.284 ms 0.251 ms
2 routera-fxp0.englab.mycompany.net (192.168.71.246) 0.251 ms 0.235 ms 0.200 ms
```

Meaning Each numbered row in the output indicates a routing "hop" in the path to the host. The three-time increments indicate the round-trip time (RTT) between the device and the hop, for each traceroute packet. To ensure that the OSPF network is healthy, verify the following information:

- The final hop in the list is the host you want to reach.
- The number of expected hops to the host matches the number of hops in the traceroute output. The appearance of more hops than expected in the output indicates that a network segment is likely not reachable. In this case, verify the routes with the **show ospf route** command.

For information about **show ospf route**, see "Verifying the Number of OSPF Routes" on page 211

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- OSPF Configuration Overview on page 165
- traceroute in the *Junos OS System Basics and Services Command Reference*

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 10

IS-IS

- IS-IS Overview on page 215
- Example: Configuring IS-IS on page 219
- IS-IS Designated Routers on page 224

IS-IS Overview

The Intermediate System-to-Intermediate System (IS-IS) protocol is a classless interior routing protocol developed by the International Organization for Standardization (ISO) as part of the development of the Open Systems Interconnection (OSI) protocol suite. Like OSPF routing, IS-IS uses hello packets that allow network convergence to occur quickly when network changes are detected. IS-IS uses the shortest path first (SPF) algorithm to determine routes. Using SPF, IS-IS evaluates network topology changes and determines if a full or partial route calculation is required.

This topic contains the following sections:

- IS-IS Areas on page 215
- System Identifier Mapping on page 216
- ISO Network Addresses on page 216
- IS-IS Path Selection on page 217
- Protocol Data Units on page 217

IS-IS Areas

An IS-IS network is a single autonomous system (AS), also called a *routing domain,* that consists of end systems and intermediate systems. End systems are network entities that send and receive packets. Intermediate systems (routers) send, receive, and relay (forward) packets.

IS-IS does not force the network to use a hierarchical physical topology. Instead, a single AS can be divided into two types of areas: Level 1 areas and Level 2 areas. A Level 1 area is similar to an OSPF stub area, and a Level 2 area interconnects all Level 1 areas. The router and its interfaces reside within one area, and Level 2 routers share link-state information. No IS-IS area functions strictly as a backbone.

Level 1 routers share intra-area routing information, and Level 2 routers share interarea information about IP addresses available within each area. Uniquely, IS-IS routers can

Copyright © 2012, Juniper Networks, Inc.

act as both Level 1 and Level 2 routers, sharing intra-area routes with other Level 1 routers and interarea routes with other Level 2 routers.

The propagation of link-state updates is determined by the level boundaries. All routers within a level maintain a complete link-state database of all other routers in the same level. Each router then uses the Dijkstra algorithm to determine the shortest path from the local router to other routers in the link-state database.

System Identifier Mapping

To provide assistance with debugging IS-IS, Junos OS supports dynamic mapping of ISO system identifiers to the hostname. Each router can be configured with a hostname that allows the system identifier-to-hostname mapping to be sent in a dynamic hostname type length value (TLV) in the IS-IS link-state PDU (LSP). The mapping permits an intermediate system in the routing domain to learn the ISO system identifier of another intermediate system.

ISO Network Addresses

IS-IS uses ISO network addresses. Each address identifies a point of connection to the network, such as a router interface, which is called *network service access point* (NSAP). NSAP addresses are supported on the loopback (**lo0**) interface.

An end system can have multiple NSAP addresses, which differ by the last byte called an *n-selector*. Each NSAP represents a service that is available at the node. In addition to multiple services, a single node can belong to multiple areas.

Each network entity also has a special address called a *network entity title* (NET) with an identical structure to an NSAP address but an n-selector of **00**. Most end systems and intermediate systems have one NET address, while intermediate systems participating in more than one area can have more than one NET address.

The following ISO addresses are examples of the IS-IS address format:

49.0001.00a0.c96b.c490.00

49.0001.2081.9716.9018.00

NETs take several forms, depending on your network requirements. NET addresses are hexadecimal and range from 8 octets to 20 octets in length. Generally, the format consists of an authority and format Identifier (AFI), a domain ID, an area ID, a system identifier, and a selector. The simplest format omits the domain ID and is 10 octets long. For example, the NET address **49.0001.1921.6800.1001.00** consists of the following parts:

- **49**—AFI
- **0001**—Area ID
- **1921.6800.1001**—System identifier
- **00**—Selector

The system identifier must be unique within the network. For an IP-only network, we recommend using the IP address of an interface on the router. Configuring a loopback

 Copyright © 2012, Juniper Networks, Inc.

NET address with the IP address is helpful when troubleshooting is required on the network.

The first part of the address is the area number, which is a variable number from 1 to 13 bytes. The first byte of the area number, **49**, is the authority and format indicator (AFI). The next bytes are the assigned area identifier and can be from 0 to 12 bytes. In the examples, **0001** is the area identifier.

The next 6 bytes are the system identifier and can be any 6 bytes unique throughout the entire domain. The system identifier is commonly the media access control (MAC) address, as shown in the first example, **00a0.c96b.c490**. Otherwise, the system identifier is the IP address expressed in binary-coded decimal (BCD) format, as shown in the second example, **2081.9716.9018**, which corresponds to **208.197.169.18**. The last byte, **00**, is the n-selector.

NOTE: **The system identifier cannot be configured as 0000.0000.0000. Using all zeros as an identifier is not supported and does not form an adjacency.**

IS-IS Path Selection

Level 1 routers store information about all the subnets within an area, and choose intranetwork paths over internetwork paths. Using the area ID portion of the NET address, Level 1 routers determine which neighboring routers are Level 1 routers within the same area.

If the destination address is not within the area, Level 1 routers forward the packet to the nearest router configured as both a Level 1 and Level 2 router within the area. The Level 1 and Level 2 router forwards the packet, using the Level 2 topology, to the proper area. The destination router, which is configured as a Level 1 and Level 2 router, then determines the best path through the destination area.

Protocol Data Units

IS-IS routers use protocol data units (PDUs) to exchange information. Each protocol data unit (PDU) shares a common header.

IS-IS Hello PDU

IS-IS hello PDUs establish adjacencies with other routers and have three different formats: one for point-to-point hello packets, one for Level 1 broadcast links, and one for Level 2 broadcast links. Level 1 routers must share the same area address to form an adjacency, while Level 2 routers do not have this limitation. The request for adjacency is encoded in the Circuit type field of the PDU.

Hello PDUs have a preset length assigned to them. The IS-IS router does not resize any PDU to match the maximum transmission unit (MTU) on a router interface. Each interface supports the maximum IS-IS PDU of 1492 bytes, and hello PDUs are padded to meet the maximum value. When the hello is sent to a neighboring router, the connecting interface supports the maximum PDU size.

Copyright © 2012, Juniper Networks, Inc.

Link-State PDU

A link-state PDU (LSP) contains information about each router in the network and the connected interfaces. Also included is metric and IS-IS neighbor information. Each LSP must be refreshed periodically on the network and is acknowledged by information within a sequence number packet.

On point-to-point links, each LSP is acknowledged by a partial sequence number PDU (PSNP), but on broadcast links, a complete sequence number PDU (CSNP) is sent out over the network. Any router that finds newer LSP information in the CSNP then purges the out-of-date entry and updates the link-state database.

LSPs support variable-length subnet mask addressing.

Complete Sequence Number PDU

The complete sequence number PDU (CSNP) lists all the link-state PDUs (LSPs) in the link-state database of the local router. Contained within the CSNP is an LSP identifier, a lifetime, a sequence number, and a checksum for each entry in the database. Periodically, a CSNP is sent on both broadcast and point-to-point links to maintain a correct database. Also, the advertisement of CSNPs occurs when an adjacency is formed with another router. Like IS-IS hello PDUs, CSNPs come in two types: Level 1 and Level 2.

When a device receives a CSNP, it checks the database entries against its own local link-state database. If it detects missing information, the device requests specific LSP details using a partial sequence number PDU (PSNP).

Partial Sequence Number PDU

A partial sequence number PDU (PSNP) is used by an IS-IS router to request LSP information from a neighboring router. A PSNP can also explicitly acknowledge the receipt of an LSP on a point-to-point link. On a broadcast link, a CSNP is used as implicit knowledge. Like hello PDUs and CSNPs, the PSNP also has two types: Level 1 and Level 2.

When a device compares a CSNP to its local database and determines that an LSP is missing, the router issues a PSNP for the missing LSP, which is returned in a link-state PDU from the router sending the CSNP. The received LSP is then stored in the local database, and an acknowledgement is sent back to the originating router.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Overview on page 3
- Understanding IS-IS Designated Routers on page 224
- Example: Configuring IS-IS on page 219
- OSPF Overview on page 160

Copyright © 2012, Juniper Networks, Inc.

Example: Configuring IS-IS

This example shows how to configure IS-IS.

- Requirements on page 219
- Overview on page 219
- Configuration on page 219
- Verification on page 221

Requirements

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

Overview

In this example, you configure the 49.0001.00a0.c96b.c490.00 NET address on the **lo0** interface. Additionally, you configure the ISO family on the **ge-0/0/1** physical interface.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set security forwarding-options family iso mode packet-based
set interfaces lo0 unit 0 family iso address 49.0001.00a0.c96b.c490.00
set interfaces ge-0/0/01 unit 0 family iso address 49.0001.00a0.c96b.c490.00
set protocols isis interface all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure IS-IS:

1. Enable IS-IS if your router is in secure context.

   ```
   [edit]
   user@host# set security forwarding-options family iso mode packet-based
   ```

   NOTE: Additionally, you must configure the ISO family on all interfaces that are supporting the IS-IS protocol.

2. Create the loopback interface.

   ```
   [edit]
   user@host# edit interfaceslo0 unit 0
   ```

3. Set the NET address.

```
[edit interfaces lo0 unit 0]
user@host# set family iso address 49.0001.00a0.c96b.c490.00
```

4. Configure the physical interface.

```
[edit]
user@host# edit interfaces ge-0/0/01 unit 0
```

5. Set the NET address.

```
[edit interfaces ge-0/0/0 unit 0]
user@host# set family iso address 49.0001.00a0.c96b.c490.00
```

6. Set IS-IS.

```
[edit ]
user@host# set protocols isis interface all
```

Results From configuration mode, confirm your configuration by entering the **show interfaces** and **show protocols** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show interfaces** and **show protocols** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
  user@host# show interfaces
ge-0/0/1 {
unit 0 {
  family inet {
  ...
  }
  family iso {
  address 49.0001.00a0.c96b.c490.00;
  }
  }
}
lo0 {
  unit 0 {
  family inet {
  ...
  }
  family iso {
  address 49.0001.00a0.c96b.c490.00;
  }
  }
}
[edit]
user@host# show protocols
  isis {
  interface all;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

 Copyright © 2012, Juniper Networks, Inc.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying IS-IS Interface Configuration on page 221
- Verifying IS-IS Interface Configuration in Detail on page 221
- Verifying IS-IS Adjacencies on page 222
- Verifying IS-IS Adjacencies in Detail on page 222

Verifying IS-IS Interface Configuration

Purpose Verify the status of the IS-IS-enabled interfaces.

Action From operational mode, enter the **show isis interface brief** command.

```
user@host> show isis interface brief
IS-IS interface database:
Interface  L CirID Level 1 DR Level 2 DR
lo0.0      3  0x1  router1  router.01
ge-0/0/1.0 2  0x9  Disabled  router.03
ge-1/0/0.0 2 0x7  Disabled  router.05
```

Meaning Verify that the output shows the intended configuration of the interfaces on which IS-IS is enabled.

Verifying IS-IS Interface Configuration in Detail

Purpose Verify the details of IS-IS-enabled interfaces.

Action From operational mode, enter the **show isis interface detail** command.

```
user@host> show isis interface detail
lo0.0
   Index:3, State:0x7, Circuit id: 0x1, Circuit type:3
   LSP interval: 100 ms, Sysid: router1
   Level Adjacencies Priority Metric Hello(s) Hold(s)
       1           0       64      0       9    27
       2           0       64      0       9    27
  ge-0/0/1.0
       Index:3, State:0x106, Circuit id: 0x9, Circuit type:2
   LSP interval: 100 ms, Sysid: router1
   Level Adjacencies Priority Metric Hello(s) Hold(s)
       1           0       64      0       9    27
       2           0       64      0       9    27
```

Meaning Check the following output fields and verify that the output shows the intended configuration of IS-IS-enabled interfaces:

- **Interface**—Interface configured for IS-IS.
- **State**—Internal implementation information.
- **Circuit id**—Circuit identifier.
- **Circuit type**—Configured level of IS-IS:

  - **1**—Level 1 only
  - **2**—Level 2 only
  - **3**—Level 1 and Level 2
- **LSP interval**—Time between IS-IS information messages.
- **Sysid**—System identifier.
- **L or Level**—Type of adjacency:
  - **1**—Level 1 only
  - **2**—Level 2 only
  - **3**—Level 1 and Level 2
- **Adjacencies**—Adjacencies established on the interface.
- **Priority**—Priority value established on the interface.
- **Metric**—Metric value for the interface.
- **Hello(s)**—Intervals between hello PDUs.
- **Hold(s)**—Hold time on the interface.

Verifying IS-IS Adjacencies

Purpose Display brief information about IS-IS neighbors.

Action From operational mode, enter the **show isis adjacency brief** command.

```
user@host> show isis adjacency brief
IS-IS adjacency database:
   Interface  System    L  State    Hold (secs) SNPA
   ge-0/0/0.0  1921.6800.5067  2  Up        13
   ge-0/0/1.0  1921.6800.5067  2 Up        25
   ge-0/0/2.0  1921.6800.5067  2  Up        19
```

Meaning Verify the adjacent routers in the IS-IS database.

Verifying IS-IS Adjacencies in Detail

Purpose Display extensive information about IS-IS neighbors.

Action From operational mode, enter the **show isis adjacency extensive** command.

```
user@host> show isis adjacency extensive
R1
  Interface: so-0/0/0.0, Level: 2, State: Up, Expires in 25 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 4w6d 19:38:52 ago
  Circuit type: 2, Speaks: IP, IPv6
  Topologies: Unicast
  Restart capable: Yes
  IP addresses: 10.1.12.1
  Transition log:
```

 Copyright © 2012, Juniper Networks, Inc.

```
  When                  State        Reason
  Wed Jul 13 16:26:11   Up           Seenself

R3
  Interface: so-0/0/1.0, Level: 2, State: Up, Expires in 23 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 6w5d 19:07:16 ago
  Circuit type: 2, Speaks: IP, IPv6
  Topologies: Unicast
  Restart capable: Yes
  IP addresses: 10.1.23.2
  Transition log:
  When                  State        Reason
  Thu Jun 30 16:57:46   Up           Seenself

R6
  Interface: so-0/0/2.0, Level: 2, State: Up, Expires in 25 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 6w0d 18:01:18 ago
  Circuit type: 2, Speaks: IP, IPv6
  Topologies: Unicast
  Restart capable: Yes
  IP addresses: 10.1.26.2
  Transition log:
  When                  State        Reason
  Tue Jul  5 18:03:45   Up           Seenself
```

Meaning Check the following fields and verify the adjacency information about IS-IS neighbors:

- **Interface**—Interface through which the neighbor is reachable.
- **L or Level**—Configured level of IS-IS:
  - **1**—Level 1 only
  - **2**—Level 2 only
  - **3**—Level 1 and Level 2

  An exclamation point before the level number indicates that the adjacency is missing an IP address.
- **State**—Status of the adjacency: **Up**, **Down**, **New**, **One-way**, **Initializing**, or **Rejected**.
- **Event**—Message that identifies the cause of a state.
- **Down reason**—Reason the adjacency is down.
- **Restart capable**—A neighbor is configured for graceful restart.
- **Transition log**—List of transitions including **When**, **State**, and **Reason**.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Configuring Designated Router Election Priority for IS-IS on page 224
- **show isis interface** in the *Junos OS Routing Protocols and Policies Command Reference*
- **show isis adjacency** in the *Junos OS Routing Protocols and Policies Command Reference*

IS-IS Designated Routers

- Understanding IS-IS Designated Routers on page 224
- Configuring Designated Router Election Priority for IS-IS on page 224

Understanding IS-IS Designated Routers

A router advertises its priority to become a designated router in its hello packets. On all multiaccess networks (physical networks that support the attachment of more than two routers, such as Ethernet networks), IS-IS uses the advertised priorities to elect a designated router for the network. This router is responsible for sending network link-state advertisements, which describe all the routers attached to the network. These advertisements are flooded throughout a single area. The priority value is meaningful only on a multiaccess network. It has no meaning on a point-to-point interface.

A router's priority for becoming the designated router is indicated by an arbitrary number from 0 through 127, which you configure on the IS-IS interface. The router with the highest priority becomes the designated router for the area (Level 1, Level 2, or both), also configured on the IS-IS interface. If routers in the network have the same priority, then the router with the highest MAC address is elected as the designated router. By default, routers have a priority value of 64.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IS-IS Overview on page 215
- Configuring Designated Router Election Priority for IS-IS on page 224

Configuring Designated Router Election Priority for IS-IS

This example shows how to configure the designated router election priority for IS-IS.

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Enable IS-IS on all interfaces within the network on which IS-IS traffic is to travel. See "Example: Configuring IS-IS" on page 219.

In this example, you configure the priority for logical interface ge-0/0/1.0 to be 100 and the level number to be 1. If this interface has the highest priority value, the router becomes the designated router for the level 1 area.

To configure a designated router election priority for IS-IS:

```
[edit]
user@host# set protocols isis interface ge-0/0/1.0 level 1 priority 100
```

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding IS-IS Designated Routers on page 224

Copyright © 2012, Juniper Networks, Inc.

- Example: Configuring IS-IS on page 219

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 11

BGP

- Understanding BGP on page 228
- BGP Routes Overview on page 229
- BGP Messages Overview on page 230
- BGP Configuration Overview on page 232
- BGP Local Preference on page 233
- BGP Peering Sessions on page 246
- BGP Path Selections on page 265
- BGP Multiple Exit Discriminator on page 300
- BGP Route Reflectors on page 304
- BGP Confederations on page 321

Copyright © 2012, Juniper Networks, Inc.

Understanding BGP

BGP is an exterior gateway protocol (EGP) that is used to exchange routing information among routers in different autonomous systems (ASs). BGP routing information includes the complete route to each destination. BGP uses the routing information to maintain a database of network reachability information, which it exchanges with other BGP systems. BGP uses the network reachability information to construct a graph of AS connectivity, which enables BGP to remove routing loops and enforce policy decisions at the AS level.

Multiprotocol BGP (MBGP) extensions enable BGP to support IP version 6 (IPv6). MBGP defines the attributes MP_REACH_NLRI and MP_UNREACH_NLRI, which are used to carry IPv6 reachability information. Network layer reachability information (NLRI) update messages carry IPv6 address prefixes of feasible routes.

BGP allows for policy-based routing. You can use routing policies to choose among multiple paths to a destination and to control the redistribution of routing information.

BGP uses TCP as its transport protocol, using port 179 for establishing connections. Running over a reliable transport protocol eliminates the need for BGP to implement update fragmentation, retransmission, acknowledgment, and sequencing.

The Junos OS routing protocol software supports BGP version 4. This version of BGP adds support for Classless Interdomain Routing (CIDR), which eliminates the concept of network classes. Instead of assuming which bits of an address represent the network by looking at the first octet, CIDR allows you to explicitly specify the number of bits in the network address, thus providing a means to decrease the size of the routing tables. BGP version 4 also supports aggregation of routes, including the aggregation of AS paths.

This section discusses the following topics:

- Autonomous Systems on page 228
- AS Paths and Attributes on page 228
- External and Internal BGP on page 229

Autonomous Systems

An *autonomous system* (AS) is a set of routers that are under a single technical administration and normally use a single interior gateway protocol and a common set of metrics to propagate routing information within the set of routers. To other ASs, an AS appears to have a single, coherent interior routing plan and presents a consistent picture of what destinations are reachable through it.

AS Paths and Attributes

The routing information that BGP systems exchange includes the complete route to each destination, as well as additional information about the route. The route to each destination is called the *AS path*, and the additional route information is included in *path attributes*. BGP uses the AS path and the path attributes to completely determine the network topology. Once BGP understands the topology, it can detect and eliminate

Copyright © 2012, Juniper Networks, Inc.

routing loops and select among groups of routes to enforce administrative preferences and routing policy decisions.

External and Internal BGP

BGP supports two types of exchanges of routing information: exchanges among different ASs and exchanges within a single AS. When used among ASs, BGP is called *external BGP* (EBGP) and BGP sessions perform *inter-AS routing*. When used within an AS, BGP is called *internal BGP* (IBGP) and BGP sessions perform *intra-AS routing*. Figure 36 on page 229 illustrates ASs, IBGP, and EBGP.

Figure 36: ASs, EBGP, and IBGP

A BGP system shares network reachability information with adjacent BGP systems, which are referred to as *neighbors* or *peers*.

BGP systems are arranged into *groups*. In an IBGP group, all peers in the group—called *internal peers*—are in the same AS. Internal peers can be anywhere in the local AS and do not have to be directly connected to one another. Internal groups use routes from an IGP to resolve forwarding addresses. They also propagate external routes among all other internal routers running IBGP, computing the next hop by taking the BGP next hop received with the route and resolving it using information from one of the interior gateway protocols.

In an EBGP group, the peers in the group—called *external peers*—are in different ASs and normally share a subnet. In an external group, the next hop is computed with respect to the interface that is shared between the external peer and the local router.

Related Documentation

- BGP Routes Overview on page 229
- BGP Messages Overview on page 230

BGP Routes Overview

A BGP route is a destination, described as an IP address prefix, and information that describes the path to the destination.

The following information describes the path:

- AS path, which is a list of numbers of the ASs that a route passes through to reach the local router. The first number in the path is that of the last AS in the path—the AS closest to the local router. The last number in the path is the AS farthest from the local router, which is generally the origin of the path.
- Path attributes, which contain additional information about the AS path that is used in routing policy.

BGP peers advertise routes to each other in update messages.

BGP stores its routes in the Junos OS routing table (**inet.0**). The routing table stores the following information about BGP routes:

- Routing information learned from update messages received from peers
- Local routing information that BGP applies to routes because of local policies
- Information that BGP advertises to BGP peers in update messages

For each prefix in the routing table, the routing protocol process selects a single best path, called the active path. Unless you configure BGP to advertise multiple paths to the same destination, BGP advertises only the active path.

The BGP router that first advertises a route assigns it one of the following values to identify its origin. During route selection, the lowest origin value is preferred.

- **0**—The router originally learned the route through an IGP (OSPF, IS-IS, or a static route).
- **1**—The router originally learned the route through an EGP (most likely BGP).
- **2**—The route's origin is unknown.

Related Documentation

- Understanding BGP Path Selection on page 265
- Example: Advertising Multiple Paths in BGP on page 269

BGP Messages Overview

All BGP messages have the same fixed-size header, which contains a marker field that is used for both synchronization and authentication, a length field that indicates the length of the packet, and a type field that indicates the message type (for example, open, update, notification, keepalive, and so on).

This section discusses the following topics:

- Open Messages on page 231
- Update Messages on page 231
- Keepalive Messages on page 232
- Notification Messages on page 232

 Copyright © 2012, Juniper Networks, Inc.

Open Messages

After a TCP connection is established between two BGP systems, they exchange BGP open messages to create a BGP connection between them. Once the connection is established, the two systems can exchange BGP messages and data traffic.

Open messages consist of the BGP header plus the following fields:

- Version—The current BGP version number is 4.
- Local AS number—You configure this by including the **autonomous-system** statement at the **[edit routing-options]** or **[edit logical-systems *logical-system-name* routing-options]** hierarchy level, as described in Specifying the Local Routing Device's AS Number.
- Hold time—Proposed hold-time value. You configure the local hold time with the BGP **hold-time** statement, as described in Configuring the Delay Before BGP Peers Mark the Routing Device as Down.
- BGP identifier—IP address of the BGP system. This address is determined when the system starts and is the same for every local interface and every BGP peer. You can configure the BGP identifier by including the **router-id** statement at the **[edit routing-options]** or **[edit logical-systems *logical-system-name* routing-options]** hierarchy level, as described in Assigning a BGP Identifier. By default, BGP uses the IP address of the first interface it finds in the router.
- Parameter field length and the parameter itself—These are optional fields.

Update Messages

BGP systems send update messages to exchange network reachability information. BGP systems use this information to construct a graph that describes the relationships among all known ASs.

Update messages consist of the BGP header plus the following optional fields:

- Unfeasible routes length—Length of the withdrawn routes field
- Withdrawn routes—IP address prefixes for the routes being withdrawn from service because they are no longer deemed reachable
- Total path attribute length—Length of the path attributes field; it lists the path attributes for a feasible route to a destination
- Path attributes—Properties of the routes, including the path origin, the multiple exit discriminator (MED), the originating system's preference for the route, and information about aggregation, communities, confederations, and route reflection
- Network layer reachability information (NLRI)—IP address prefixes of feasible routes being advertised in the update message

Copyright © 2012, Juniper Networks, Inc.

Keepalive Messages

BGP systems exchange keepalive messages to determine whether a link or host has failed or is no longer available. Keepalive messages are exchanged often enough so that the hold timer does not expire. These messages consist only of the BGP header.

Notification Messages

BGP systems send notification messages when an error condition is detected. After the message is sent, the BGP session and the TCP connection between the BGP systems are closed. Notification messages consist of the BGP header plus the error code and subcode, and data that describes the error.

Related Documentation

- Understanding BGP on page 228
- BGP Routes Overview on page 229

BGP Configuration Overview

To configure the device as a node in a BGP network:

1. Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Configure point-to-point peering sessions. See "Example: Configuring External BGP Point-to-Point Peer Sessions" on page 247.
3. Configure IBGP sessions between peers. See "Example: Configuring Internal BGP Peer Sessions" on page 254.
4. Configure a routing policy to advertise the BGP routes.
5. (Optional) Configure route reflector clusters. See "Example: Configuring a Route Reflector" on page 306.
6. (Optional) Subdivide autonomous systems (ASs). See "Example: Configuring BGP Confederations" on page 322.
7. (Optional) Assign a router ID to each routing device running BGP.
8. (Optional) Configure a local preference to direct all outbound AS traffic to a specific peer. See "Example: Configuring the Local Preference Value for BGP Routes" on page 233 in the *Junos OS Routing Protocols Configuration Guide*.
9. (Optional) Configure routing table path selection options that define different ways to compare multiple exit discriminators (MEDs). See Configuring Routing Table Path Selection for BGP in the *Junos OS Routing Protocols Configuration Guide*.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding BGP on page 228
- Configuring BGP in the *Junos OS Routing Protocols Configuration Guide*
- Minimum BGP Configuration in the *Junos OS Routing Protocols Configuration Guide*

 Copyright © 2012, Juniper Networks, Inc.

BGP Local Preference

- Understanding the BGP Local Preference on page 233
- Example: Configuring the Local Preference Value for BGP Routes on page 233

Understanding the BGP Local Preference

Internal BGP (IBGP) sessions use a metric called the *local preference*, which is carried in IBGP update packets in the path attribute **LOCAL_PREF**. When an autonomous system (AS) has multiple routes to another AS, the local preference indicates the degree of preference for one route over the other routes. The route with the highest local preference value is preferred.

The **LOCAL_PREF** path attribute is always advertised to IBGP peers and to neighboring confederations. It is never advertised to external BGP (EBGP) peers. The default behavior is to not modify the **LOCAL_PREF** path attribute if it is present.

The **LOCAL_PREF** path attribute applies at export time only, when the routes are exported from the routing table into BGP.

If a BGP route is received without a **LOCAL_PREF** attribute, the route is stored in the routing table and advertised by BGP as if it were received with a **LOCAL_PREF** value of 100. A non-BGP route that is advertised by BGP is advertised with a **LOCAL_PREF** value of 100 by default.

Related Documentation

- Example: Configuring the Local Preference Value for BGP Routes on page 233

Example: Configuring the Local Preference Value for BGP Routes

This example shows how to configure local preference in internal BGP (IBGP) peer sessions.

- Requirements on page 233
- Overview on page 233
- Configuration on page 234
- Verification on page 244

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

To change the local preference metric advertised in the path attribute, you must include the **local-preference** statement, specifying a value from 0 through 4,294,967,295 ($2^{32} - 1$).

There are several reasons you might want to prefer one path over another. For example, compared to other paths, one path might be less expensive to use, might have higher bandwidth, or might be more stable.

Figure 37 on page 234 shows a typical network with internal peer sessions and multiple exit points to a neighboring AS.

Figure 37: Typical Network with IBGP Sessions and Multiple Exit Points

To reach Device R4, Device R1 can take a path through either Device R2 or Device R3. By default, the local preference is 100 for either route. When the local preferences are equal, Junos OS has rules for breaking the tie and choosing a path. (See "Understanding BGP Path Selection" on page 265.) In this example, the active route is through Device R2 because the router ID of Device R2 is lower than the router ID of Device R3. The following example shows how to override the default behavior with an explicit setting for the local preference. The example configures a local preference of 300 on Device R3, thereby making Device R3 the preferred path to reach Device R4.

Configuration

- Configuring Device R1 on page 236
- Configuring Device R2 on page 238
- Configuring Device R3 on page 240
- Configuring Device R4 on page 242

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 12.12.12.1/24
set interfaces fe-1/2/1 unit 2 family inet address 13.13.13.1/24
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.1.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
```


Copyright © 2012, Juniper Networks, Inc.

```
set protocols ospf area 0.0.0.0 interface fe-1/2/1.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.1.1
```

Device R2

```
set interfaces fe-1/2/0 unit 3 family inet address 12.12.12.2/24
set interfaces fe-1/2/1 unit 4 family inet address 24.24.24.2/24
set interfaces lo0 unit 2 family inet address 192.168.2.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.2.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 24.24.24.4
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.2.1
```

Device R3

```
set interfaces fe-1/2/0 unit 5 family inet address 13.13.13.3/24
set interfaces fe-1/2/1 unit 6 family inet address 34.34.34.3/24
set interfaces lo0 unit 3 family inet address 192.168.3.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.3.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 34.34.34.4
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.5
set protocols ospf area 0.0.0.0 interface fe-1/2/1.6
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.3.1
```

Device R4

```
set interfaces fe-1/2/0 unit 7 family inet address 24.24.24.4/24
set interfaces fe-1/2/1 unit 8 family inet address 34.34.34.4/24
set interfaces lo0 unit 4 family inet address 192.168.4.1/32
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 123
set protocols bgp group external neighbor 34.34.34.3
set protocols bgp group external neighbor 24.24.24.2
set policy-options policy-statement send-direct term 1 from protocol direct
```

Copyright © 2012, Juniper Networks, Inc.

```
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 4
set routing-options router-id 192.168.4.1
```

Configuring Device R1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device R1:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 1]
user@R1# set family inet address 12.12.12.1/24

[edit interfaces fe-1/2/1 unit 2]
user@R1# set family inet address 13.13.13.1/24

[edit interfaces lo0 unit 1]
user@R1# set family inet address 192.168.1.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R1# set type internal
user@R1# set local-address 192.168.1.1
user@R1# set export send-direct
user@R1# set neighbor 192.168.2.1
user@R1# set neighbor 192.168.3.1
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface lo0.1 passive
user@R1# set interface fe-1/2/0.1
user@R1# set interface fe-1/2/1.2
```

4. Configure a policy that accepts direct routes.

NOTE: Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R1# set from protocol direct
user@R1# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 123
user@R1# set router-id 192.168.1.1
```


Copyright © 2012, Juniper Networks, Inc.

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show interfaces
fe-1/2/0 {
   unit 1 {
      family inet {
         address 12.12.12.1/24;
      }
   }
}
fe-1/2/1 {
   unit 2 {
      family inet {
         address 13.13.13.1/24;
      }
   }
}
lo0 {
   unit 1 {
      family inet {
         address 192.168.1.1/32;
      }
   }
}
```

```
user@R1# show policy-options
policy-statement send-direct {
   term 1 {
      from protocol direct;
      then accept;
   }
}
```

```
user@R1# show protocols
bgp {
   group internal {
      type internal;
      local-address 192.168.1.1;
      export send-direct;
      neighbor 192.168.2.1;
      neighbor 192.168.3.1;
   }
}
ospf {
   area 0.0.0.0 {
      interface lo0.1 {
         passive;
      }
      interface fe-1/2/0.1;
      interface fe-1/2/1.2;
   }
}
```

```
user@R1# show routing-options
```

```
autonomous-system 123;
router-id 192.168.1.1;
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device R2

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device R2:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 3]
user@R2# set family inet address 12.12.12.21/24

[edit interfaces fe-1/2/1 unit 4]
user@R2# set family inet address 24.24.24.2/24

[edit interfaces lo0 unit 2]
user@R2# set family inet address 192.168.2.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R2# set type internal
user@R2# set local-address 192.168.2.1
user@R2# set export send-direct
user@R2# set neighbor 192.168.1.1
user@R2# set neighbor 192.168.3.1

[edit protocols bgp group external]
user@R2# set type external
user@R2# set export send-direct
user@R2# set peer-as 4
user@R2# set neighbor 24.24.24.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R2# set interface lo0.2 passive
user@R2# set interface fe-1/2/0.3
user@R2# set interface fe-1/2/1.4
```

4. Configure a policy that accepts direct routes.

NOTE: Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R2# set from protocol direct
user@R2# set then accept
```


Copyright © 2012, Juniper Networks, Inc.

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R2# set autonomous-system 123
user@R2# set router-id 192.168.2.1
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R2# show interfaces
fe-1/2/0 {
   unit 3 {
      family inet {
         address 12.12.12.2/24;
      }
   }
}
fe-1/2/1 {
   unit 4 {
      family inet {
         address 24.24.24.2/24;
      }
   }
}
lo0 {
   unit 2 {
      family inet {
         address 192.168.2.1/32;
      }
   }
}
```

```
user@R2# show policy-options
policy-statement send-direct {
   term 1 {
      from protocol direct;
      then accept;
   }
}
```

```
user@R2# show protocols
bgp {
   group internal {
      type internal;
      local-address 192.168.2.1;
      export send-direct;
      neighbor 192.168.1.1;
      neighbor 192.168.3.1;
   }
   group external {
      type external;
      export send-direct;
      peer-as 4;
      neighbor 24.24.24.4;
```

```
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.2 {
        passive;
      }
      interface fe-1/2/0.3;
      interface fe-1/2/1.4;
    }
  }
```

```
user@R2# show routing-options
autonomous-system 123;
router-id 192.168.2.1;
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device R3

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device R3:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 5]
user@R3# set family inet address 13.13.13.3/24

[edit interfaces fe-1/2/1 unit 6]
user@R3# set family inet address 34.34.34.3/24

[edit interfaces lo0 unit 3]
user@R3# set family inet address 192.168.3.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R3# set type internal
user@R3# set local-address 192.168.3.1
user@R3# set export send-direct
user@R3# set neighbor 192.168.1.1
user@R3# set neighbor 192.168.2.1

[edit protocols bgp group external]
user@R3# set type external
user@R3# set export send-direct
user@R3# set peer-as 4
user@R3# set neighbor 34.34.34.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R3# set interface lo0.3 passive
user@R3# set interface fe-1/2/0.5
```


Copyright © 2012, Juniper Networks, Inc.

```
user@R3# set interface fe-1/2/1.6
```

4. Configure a policy that accepts direct routes.

NOTE: Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R3# set from protocol direct
user@R3# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R3# set autonomous-system 123
user@R3# set router-id 192.168.3.1
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R3# show interfaces
fe-1/2/0 {
   unit 5 {
      family inet {
         address 13.13.13.3/24;
      }
   }
}
fe-1/2/1 {
   unit 6 {
      family inet {
         address 34.34.34.3/24;
      }
   }
}
lo0 {
   unit 3 {
      family inet {
         address 192.168.3.1/32;
      }
   }
}
```

```
user@R3# show policy-options
policy-statement send-direct {
   term 1 {
      from protocol direct;
      then accept;
   }
}
```

```
user@R3# show protocols
```

```
bgp {
   group internal {
      type internal;
      local-address 192.168.3.1;
      export send-direct;
      neighbor 192.168.1.1;
      neighbor 192.168.2.1;
   }
   group external {
      type external;
      export send-direct;
      peer-as 4;
      neighbor 34.34.34.4;
   }
}
ospf {
   area 0.0.0.0 {
      interface lo0.3 {
         passive;
      }
      interface fe-1/2/0.5;
      interface fe-1/2/1.6;
   }
}
```

```
user@R3# show routing-options
autonomous-system 123;
router-id 192.168.3.1;
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device R4

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device R4:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 7]
user@R4# set family inet address 24.24.24.4/24

[edit interfaces fe-1/2/1 unit 8]
user@R4# set family inet address 34.34.34.4/24

[edit interfaces lo0 unit 4]
user@R4# set family inet address 192.168.4.1/32
```

2. Configure BGP.

```
[edit protocols bgp group external]
user@R4# set type external
user@R4# set export send-direct
user@R4# set peer-as 123
```


Copyright © 2012, Juniper Networks, Inc.

```
user@R4# set neighbor 34.34.34.3
user@R4# set neighbor 24.24.24.2
```

3. Configure a policy that accepts direct routes.

NOTE: Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R4# set from protocol direct
user@R4# set then accept
```

4. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R4# set autonomous-system 4
user@R4# set router-id 192.168.4.1
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R4# show interfaces
fe-1/2/0 {
   unit 7 {
      family inet {
         address 24.24.24.4/24;
      }
   }
}
fe-1/2/1 {
   unit 8 {
      family inet {
         address 34.34.34.4/24;
      }
   }
}
lo0 {
   unit 4 {
      family inet {
         address 192.168.4.1/32;
      }
   }
}
```

```
user@R4# show policy-options
policy-statement send-direct {
   term 1 {
      from protocol direct;
      then accept;
   }
}
```

```
user@R4# show protocols
bgp {
  group external {
    type external;
    export send-direct;
    peer-as 123;
    neighbor 34.34.34.3;
    neighbor 24.24.24.2;
  }
}

user@R4# show routing-options
autonomous-system 4;
router-id 192.168.4.1;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Checking the Active Path From Device R1 to Device R4 on page 244
- Altering the Local Preference to Change the Path Selection on page 245
- Rechecking the Active Path From Device R1 to Device R4 on page 245

Checking the Active Path From Device R1 to Device R4

Purpose Verify that the active path from Device R1 to Device R4 goes through Device R2.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@R1> show route protocol bgp
inet.0: 11 destinations, 18 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

12.12.12.0/24       [BGP/170] 00:11:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
13.13.13.0/24       [BGP/170] 00:11:48, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
24.24.24.0/24       [BGP/170] 00:11:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
34.34.34.0/24       [BGP/170] 00:11:48, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.2.1/32      [BGP/170] 00:11:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
192.168.3.1/32      [BGP/170] 00:11:48, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.4.1/32     *[BGP/170] 00:05:14, localpref 100, from 192.168.2.1
                      AS path: 4 I
                    > to 12.12.12.2 via fe-1/2/0.1
                    [BGP/170] 00:05:14, localpref 100, from 192.168.3.1
```

 Copyright © 2012, Juniper Networks, Inc.

```
                    AS path: 4 I
                  > to 13.13.13.3 via fe-1/2/1.2
```

Meaning The asterisk (*) shows that the preferred path is through Device R2. In the default configuration, Device R2 has a lower router ID than Device R3. The router ID is controlling the path selection.

Altering the Local Preference to Change the Path Selection

Purpose Change the path so that it goes through Device R3.

Action From configuration mode, enter the **set local-preference 300** command.

```
[edit protocols bgp group internal]
user@R1# set local-preference 300
user@R1# commit
```

Rechecking the Active Path From Device R1 to Device R4

Purpose Verify that the active path from Device R1 to Device R4 goes through Device R3.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@R1> show route protocol bgp
inet.0: 11 destinations, 17 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

12.12.12.0/24       [BGP/170] 00:16:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
13.13.13.0/24       [BGP/170] 00:00:22, localpref 300, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
24.24.24.0/24       [BGP/170] 00:16:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
34.34.34.0/24       [BGP/170] 00:00:22, localpref 300, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.2.1/32      [BGP/170] 00:16:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
192.168.3.1/32      [BGP/170] 00:00:22, localpref 300, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.4.1/32     *[BGP/170] 00:00:21, localpref 300, from 192.168.3.1
                      AS path: 4 I
                    > to 13.13.13.3 via fe-1/2/1.2
```

Meaning The asterisk (*) shows that the preferred path is through Device R3. In the altered configuration, Device R3 has a higher local preference than Device R2. The local preference is controlling the path selection.

Related Documentation

- Understanding the BGP Local Preference on page 233
- BGP Configuration Overview on page 232

BGP Peering Sessions

- Understanding External BGP Peering Sessions on page 246
- Example: Configuring External BGP Point-to-Point Peer Sessions on page 247
- Example: Configuring Internal BGP Peer Sessions on page 254

Understanding External BGP Peering Sessions

To establish point-to-point connections between peer autonomous systems (ASs), you configure a BGP session on each interface of a point-to-point link. Generally, such sessions are made at network exit points with neighboring hosts outside the AS. Figure 38 on page 246 shows an example of a BGP peering session.

Figure 38: BGP Peering Session

In Figure 38 on page 246, Router A is a gateway router for AS 3, and Router B is a gateway router for AS 10. For traffic internal to either AS, an interior gateway protocol (IGP) is used (OSPF, for instance). To route traffic between peer ASs, a BGP session is used.

You arrange BGP routing devices into groups of peers. Different peer groups must have different group types, AS numbers, or route reflector cluster identifiers.

To define a BGP group that recognizes only the specified BGP systems as peers, statically configure all the system's peers by including one or more **neighbor** statements. The peer neighbor's address can be either an IPv6 or IPv4 address.

As the number of external BGP (EBGP) groups increases, the ability to support a large number of BGP sessions might become a scaling issue. The preferred way to configure a large number of BGP neighbors is to configure a few groups consisting of multiple neighbors per group. Supporting fewer EBGP groups generally scales better than supporting a large number of EBGP groups. This becomes more evident in the case of hundreds of EBGP groups when compared with a few EBGP groups with multiple peers in each group.

After the BGP peers are established, BGP routes are not automatically advertised by the BGP peers. At each BGP-enabled device, policy configuration is required to export the local, static, or IGP-learned routes into the BGP RIB and then advertise them as BGP routes to the other peers. BGP's advertisement policy, by default, does not advertise any non-BGP routes (such as local routes) to peers.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding BGP on page 228
- Example: Configuring External BGP Point-to-Point Peer Sessions on page 247
- Example: Configuring Internal BGP Peer Sessions on page 254

Example: Configuring External BGP Point-to-Point Peer Sessions

This example shows how to configure BGP point-to-point peer sessions.

- Requirements on page 247
- Overview on page 247
- Configuration on page 248
- Verification on page 250

Requirements

Before you begin, if the default BGP policy is not adequate for your network, configure routing policies to filter incoming BGP routes and to advertise BGP routes.

Overview

Figure 39 on page 247 shows a network with BGP peer sessions. In the sample network, Device E in AS 17 has BGP peer sessions to a group of peers called **external-peers**. Peers A, B, and C reside in AS 22 and have IP addresses 10.10.10.2, 10.10.10.6, and 10.10.10.10. Peer D resides in AS 79, at IP address 10.21.7.2. This example shows the configuration on Device E.

Figure 39: Typical Network with BGP Peer Sessions

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set interfaces ge-1/2/0 unit 0 description to-A
set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/30
set interfaces ge-0/0/1 unit 5 description to-B
set interfaces ge-0/0/1 unit 5 family inet address 10.10.10.5/30
set interfaces ge-0/1/0 unit 9 description to-C
set interfaces ge-0/1/0 unit 9 family inet address 10.10.10.9/30
set interfaces ge-1/2/1 unit 21 description to-D
set interfaces ge-1/2/1 unit 21 family inet address 10.21.7.1/30
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 22
set protocols bgp group external-peers neighbor 10.10.10.2
set protocols bgp group external-peers neighbor 10.10.10.6
set protocols bgp group external-peers neighbor 10.10.10.10
set protocols bgp group external-peers neighbor 10.21.7.2 peer-as 79
set routing-options autonomous-system 17
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure the BGP peer sessions:

1. Configure the interfaces to Peers A, B, C, and D.

```
[edit interfaces]
user@E# set ge-1/2/0 unit 0 description to-A
user@E# set ge-1/2/0 unit 0 family inet address 10.10.10.1/30
user@E# set ge-0/0/1 unit 5 description to-B
user@E# set ge-0/0/1 unit 5 family inet address 10.10.10.5/30
user@E# set ge-0/1/0 unit 9 description to-C
user@E# set ge-0/1/0 unit 9 family inet address 10.10.10.9/30
user@E# set ge-1/2/1 unit 21 description to-D
user@E# set ge-1/2/1 unit 21 family inet address 10.21.7.1/30
```

2. Set the autonomous system (AS) number.

```
[edit routing-options]
user@E# set autonomous-system 17
```

3. Create the BGP group, and add the external neighbor addresses.

```
[edit protocols bgp group external-peers]
user@E# set neighbor 10.10.10.2
user@E# set neighbor 10.10.10.6
user@E# set neighbor 10.10.10.10
```

4. Specify the autonomous system (AS) number of the external AS.

```
[edit protocols bgp group external-peers]
user@E# set peer-as 22
```


Copyright © 2012, Juniper Networks, Inc.

5. Add Peer D, and set the AS number at the individual neighbor level.

```
[edit protocols bgp group external-peers]
user@E# set neighbor 10.21.7.2 peer-as 79
```

6. Set the peer type to external BGP (EBGP).

```
[edit protocols bgp group external-peers]
user@E# set type external
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@E# show interfaces
ge-1/2/0 {
   unit 0 {
      description to-A;
      family inet {
         address 10.10.10.1/30;
      }
   }
}
ge-0/0/1 {
   unit 5 {
      description to-B;
      family inet {
         address 10.10.10.5/30;
      }
   }
}
ge-0/1/0 {
   unit 9 {
      description to-C;
      family inet {
         address 10.10.10.9/30;
      }
   }
}
ge-1/2/1 {
   unit 21 {
      description to-D;
      family inet {
         address 10.21.7.1/30;
      }
   }
}
```

```
[edit]
user@E# show protocols
bgp {
   group external-peers {
      type external;
      peer-as 22;
      neighbor 10.10.10.2;
```

```
        neighbor 10.10.10.6;
        neighbor 10.10.10.10;
        neighbor 10.21.7.2 {
          peer-as 79;
        }
      }
    }

[edit]
user@E# show routing-options
autonomous-system 17;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying BGP Neighbors on page 250
- Verifying BGP Groups on page 253
- Verifying BGP Summary Information on page 253
- Verifying Reachability of All Peers in a BGP Network on page 253

Verifying BGP Neighbors

Purpose Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

Action From operational mode, run the **show bgp neighbor** command.

```
user@E> show bgp neighbor
Peer: 10.10.10.2+179 AS 22     Local: 10.10.10.1+65406 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.10.10.2       Local ID: 10.10.10.1       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  Local Interface: ge-1/2/0.0
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
```

 Copyright © 2012, Juniper Networks, Inc.

```
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 10   Sent 6    Checked 1
  Input messages:  Total 8522   Updates 1       Refreshes 0     Octets 161922
  Output messages: Total 8433   Updates 0       Refreshes 0     Octets 160290
  Output Queue[0]: 0

Peer: 10.10.10.6+54781 AS 22   Local: 10.10.10.5+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.10.10.6       Local ID: 10.10.10.1       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  Local Interface: ge-0/0/1.5
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 12   Sent 6    Checked 33
  Input messages:  Total 8527   Updates 1       Refreshes 0     Octets 162057
  Output messages: Total 8430   Updates 0       Refreshes 0     Octets 160233
  Output Queue[0]: 0

Peer: 10.10.10.10+55012 AS 22  Local: 10.10.10.9+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.10.10.10      Local ID: 10.10.10.1       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 2
  BFD: disabled, down
  Local Interface: fe-0/1/0.9
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
```

```
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 15   Sent 6    Checked 37
  Input messages:  Total 8527   Updates 1       Refreshes 0     Octets 162057
  Output messages: Total 8429   Updates 0       Refreshes 0     Octets 160214
  Output Queue[0]: 0

Peer: 10.21.7.2+61867 AS 79    Local: 10.21.7.1+179 AS 17
  Type: External    State: Established    Flags: <ImportEval Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.21.7.2        Local ID: 10.10.10.1       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 3
  BFD: disabled, down
  Local Interface: ge-1/2/1.21
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 79)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 28   Sent 24   Checked 47
  Input messages:  Total 8521   Updates 1       Refreshes 0     Octets 161943
  Output messages: Total 8427   Updates 0       Refreshes 0     Octets 160176
  Output Queue[0]: 0
```


Copyright © 2012, Juniper Networks, Inc.

Verifying BGP Groups

Purpose Verify that the BGP groups are configured correctly.

Action From operational mode, run the **show bgp group** command.

```
user@E> show bgp group
Group Type: External                                Local AS: 17
  Name: external-peers  Index: 0                    Flags: <>
  Holdtime: 0
  Total peers: 4        Established: 4
  10.10.10.2+179
  10.10.10.6+54781
  10.10.10.10+55012
  10.21.7.2+61867
  inet.0: 0/0/0/0

Groups: 1  Peers: 4    External: 4    Internal: 0    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
```

Verifying BGP Summary Information

Purpose Verify that the BGP configuration is correct.

Action From operational mode, run the **show bgp summary** command.

```
user@E> show bgp summary
Groups: 1 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.10.10.2               22       8559       8470       0       0 2d 16:12:56
0/0/0/0              0/0/0/0
10.10.10.6               22       8566       8468       0       0 2d 16:12:12
0/0/0/0              0/0/0/0
10.10.10.10              22       8565       8466       0       0 2d 16:11:31
0/0/0/0              0/0/0/0
10.21.7.2                79       8560       8465       0       0 2d 16:10:58
0/0/0/0              0/0/0/0
```

Verifying Reachability of All Peers in a BGP Network

Purpose By using the ping tool on each peer address in the network, verify that all peers in the network are reachable from each device.

Action For each device in the BGP network:

1. In the J-Web interface, select **Troubleshoot>Ping Host**.
2. In the Remote Host box, type the name of a host for which you want to verify reachability from the device.
3. Click **Start**. Output appears on a separate page.

Sample Output

```
PING 10.10.10.10 : 56 data bytes
64 bytes from 10.10.10.10: icmp_seq=0 ttl=255 time=0.382 ms
64 bytes from 10.10.10.10: icmp_seq=1 ttl=255 time=0.266 ms
```

Meaning If a host is active, it generates an ICMP response. If this response is received, the round-trip time is listed in the **time** field.

Related Documentation

- *Junos OS Policy Framework Configuration Guide*
- Understanding External BGP Peering Sessions on page 246
- Example: Configuring Internal BGP Peer Sessions on page 254
- BGP Configuration Overview on page 232

Example: Configuring Internal BGP Peer Sessions

This example shows how to configure internal BGP peer sessions.

- Requirements on page 254
- Overview on page 254
- Configuration on page 255
- Verification on page 263

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

In this example, you configure internal BGP (IBGP) peer sessions. The loopback interface (lo0) is used to establish connections between IBGP peers. The loopback interface is always up as long as the device is operating. If there is a route to the loopback address, the IBGP peer session stays up. If a physical interface address is used instead and that interface goes up and down, the IBGP peer session also goes up and down. Thus, if the device has link redundancy, the loopback interface provides fault tolerance in case the physical interface or one of the links goes down.

When a device peers with a remote device's loopback interface address, the local device expects BGP update messages to come from (be sourced by) the remote device's loopback interface address. The **local-address** statement enables you to specify the source information in BGP update messages. If you omit the **local-address** statement, the expected source of BGP update messages is based on the device's source address selection rules, which normally results in the egress interface address being the expected source of update messages. When this happens, the peer session is not established because a mismatch exists between the expected source address (the egress interface of the peer) and the actual source (the loopback interface of the peer). To make sure

 Copyright © 2012, Juniper Networks, Inc.

that the expected source address matches the actual source address, specify the loopback interface address in the **local-address** statement.

Because IBGP supports multihop connections, IBGP neighbors can be located anywhere within the autonomous system (AS) and often do not share a link. A recursive route lookup resolves the loopback peer address to an IP forwarding next hop. In this example, this service is provided by OSPF. Although interior gateway protocol (IGP) neighbors do not need to be directly connected, they do need to be fully meshed. In this case, fully meshed means that each device is logically connected to every other device through neighbor peer relationships. The **neighbor** statement creates the mesh.

NOTE: **The requirement for a full mesh is waived if you configure a confederation or route reflection.**

After the BGP peers are established, BGP routes are not automatically advertised by the BGP peers. At each BGP-enabled device, policy configuration is required to export the local, static, or IGP-learned routes into the BGP routing information base (RIB) and then advertise them as BGP routes to the other peers. BGP's advertisement policy, by default, does not advertise any non-BGP routes (such as local routes) to peers.

In the sample network, the devices in AS 17 are fully meshed in the group **internal-peers**. The devices have loopback addresses 192.168.6.5, 192.163.6.4, and 192.168.40.4.

Figure 40 on page 255 shows a typical network with internal peer sessions.

Figure 40: Typical Network with IBGP Sessions

Configuration

- Configuring Device A on page 257
- Configuring Device B on page 259
- Configuring Device C on page 261

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network

configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device A

```
set interfaces ge-0/1/0 unit 1 description to-B
set interfaces ge-0/1/0 unit 1 family inet address 10.10.10.1/30
set interfaces lo0 unit 1 family inet address 192.168.6.5/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers description “connections to B and C”
set protocols bgp group internal-peers local-address 192.168.6.5
set protocols bgp group internal-peers export send-direct
set protocols bgp group internal-peers neighbor 192.163.6.4
set protocols bgp group internal-peers neighbor 192.168.40.4
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface ge-0/1/0.1
set policy-options policy-statement send-direct term 2 from protocol direct
set policy-options policy-statement send-direct term 2 then accept
set routing-options router-id 192.168.6.5
set routing-options autonomous-system 17
```

Device B

```
set interfaces ge-0/1/0 unit 2 description to-A
set interfaces ge-0/1/0 unit 2 family inet address 10.10.10.2/30
set interfaces ge-0/1/1 unit 5 description to-C
set interfaces ge-0/1/1 unit 5 family inet address 10.10.10.5/30
set interfaces lo0 unit 2 family inet address 192.163.6.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers description “connections to A and C”
set protocols bgp group internal-peers local-address 192.163.6.4
set protocols bgp group internal-peers export send-direct
set protocols bgp group internal-peers neighbor 192.168.40.4
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface ge-0/1/0.2
set protocols ospf area 0.0.0.0 interface ge-0/1/1.5
set policy-options policy-statement send-direct term 2 from protocol direct
set policy-options policy-statement send-direct term 2 then accept
set routing-options router-id 192.163.6.4
set routing-options autonomous-system 17
```

Device C

```
set interfaces ge-0/1/0 unit 6 description to-B
set interfaces ge-0/1/0 unit 6 family inet address 10.10.10.6/30
set interfaces lo0 unit 3 family inet address 192.168.40.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers description “connections to A and B”
set protocols bgp group internal-peers local-address 192.168.40.4
set protocols bgp group internal-peers export send-direct
set protocols bgp group internal-peers neighbor 192.163.6.4
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface ge-0/1/0.6
set policy-options policy-statement send-direct term 2 from protocol direct
set policy-options policy-statement send-direct term 2 then accept
set routing-options router-id 192.168.40.4
set routing-options autonomous-system 17
```


Copyright © 2012, Juniper Networks, Inc.

Configuring Device A

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure internal BGP peer sessions on Device A:

1. Configure the interfaces.

```
[edit interfaces ge-0/1/0 unit 1]
user@A# set description to-B
user@A# set family inet address 10.10.10.1/30

[edit interfaces]
user@A# set lo0 unit 1 family inet address 192.168.6.5/32
```

2. Configure BGP.

   The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```
[edit protocols bgp group internal-peers]
user@A# set type internal
user@A# set description "connections to B and C"
user@A# set local-address 192.168.6.5
user@A# set export send-direct
user@A# set neighbor 192.163.6.4
user@A# set neighbor 192.168.40.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@A# set interface lo0.1 passive
user@A# set interface ge-0/1/0.1
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 2]
user@A# set from protocol direct
user@A# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@A# set router-id 192.168.6.5
user@A# set autonomous-system 17
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@A# show interfaces
ge-0/1/0 {
```

```
    unit 1 {
      description to-B;
      family inet {
        address 10.10.10.1/30;
      }
    }
  }
  lo0 {
    unit 1 {
      family inet {
        address 192.168.6.5/32;
      }
    }
  }
```

```
user@A# show policy-options
policy-statement send-direct {
  term 2 {
    from protocol direct;
    then accept;
  }
}
```

```
user@A# show protocols
bgp {
  group internal-peers {
    type internal;
    description “connections to B and C”;
    local-address 192.168.6.5;
    export send-direct;
    neighbor 192.163.6.4;
    neighbor 192.168.40.4;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.1 {
      passive;
    }
    interface ge-0/1/0.1;
  }
}
```

```
user@A# show routing-options
router-id 192.168.6.5;
autonomous-system 17;
```

If you are done configuring the device, enter **commit** from configuration mode.

Copyright © 2012, Juniper Networks, Inc.

Configuring Device B

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode.

To configure internal BGP peer sessions on Device B:

1. Configure the interfaces.

```
[edit interfaces ge-0/1/0 unit 2]
user@B# set description to-A
user@B# set family inet address 10.10.10.2/30

[edit interfaces ge-0/1/1]
user@B# set unit 5 description to-C
user@B# set unit 5 family inet address 10.10.10.5/30

[edit interfaces]
user@B# set lo0 unit 2 family inet address 192.163.6.4/32
```

2. Configure BGP.

   The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```
[edit protocols bgp group internal-peers]
user@B# set type internal
user@B# set description "connections to A and C"
user@B# set local-address 192.163.6.4
user@B# set export send-direct
user@B# set neighbor 192.168.40.4
user@B# set neighbor 192.168.6.5
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@B# set interface lo0.2 passive
user@B# set interface ge-0/1/0.2
user@B# set interface ge-0/1/1.5
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 2]
user@B# set from protocol direct
user@B# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@B# set router-id 192.163.6.4
user@B# set autonomous-system 17
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@B# show interfaces
ge-0/1/0 {
   unit 2 {
      description to-A;
      family inet {
         address 10.10.10.2/30;
      }
   }
}
ge-0/1/1 {
   unit 5 {
      description to-C;
      family inet {
         address 10.10.10.5/30;
      }
   }
}
lo0 {
   unit 2 {
      family inet {
         address 192.163.6.4/32;
      }
   }
}
```

```
user@B# show policy-options
policy-statement send-direct {
   term 2 {
      from protocol direct;
      then accept;
   }
}
```

```
user@B# show protocols
bgp {
   group internal-peers {
      type internal;
      description “connections to A and C”;
      local-address 192.163.6.4;
      export send-direct;
      neighbor 192.168.40.4;
      neighbor 192.168.6.5;
   }
}
ospf {
   area 0.0.0.0 {
      interface lo0.2 {
         passive;
      }
      interface ge-0/1/0.2;
      interface ge-0/1/1.5;
```

 Copyright © 2012, Juniper Networks, Inc.

```
    }
  }

  user@B# show routing-options
  router-id 192.163.6.4;
  autonomous-system 17;
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Device C

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure internal BGP peer sessions on Device C:

1. Configure the interfaces.

```
[edit interfaces ge-0/1/0 unit 6]
user@C# set description to-B
user@C# set family inet address 10.10.10.6/30

[edit interfaces]
user@C# set lo0 unit 3 family inet address 192.168.40.4/32
```

2. Configure BGP.

   The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```
[edit protocols bgp group internal-peers]
user@C# set type internal
user@C# set description "connections to A and B"
user@C# set local-address 192.168.40.4
user@C# set export send-direct
user@C# set neighbor 192.163.6.4
user@C# set neighbor 192.168.6.5
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@C# set interface lo0.3 passive
user@C# set interface ge-0/1/0.6
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 2]
user@C# set from protocol direct
user@C# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@C# set router-id 192.168.40.4
user@C# set autonomous-system 17
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@C# show interfaces
ge-0/1/0 {
   unit 6 {
      description to-B;
      family inet {
         address 10.10.10.6/30;
      }
   }
}
lo0 {
   unit 3 {
      family inet {
         address 192.168.40.4/32;
      }
   }
}
```

```
user@C# show policy-options
policy-statement send-direct {
   term 2 {
      from protocol direct;
      then accept;
   }
}
```

```
user@C# show protocols
bgp {
   group internal-peers {
      type internal;
      description “connections to A and B”;
      local-address 192.168.40.4;
      export send-direct;
      neighbor 192.163.6.4;
      neighbor 192.168.6.5;
   }
}
ospf {
   area 0.0.0.0 {
      interface lo0.3 {
         passive;
      }
      interface ge-0/1/0.6;
   }
}
```

```
user@C# show routing-options
router-id 192.168.40.4;
autonomous-system 17;
```

If you are done configuring the device, enter **commit** from configuration mode.

 Copyright © 2012, Juniper Networks, Inc.

Verification

Confirm that the configuration is working properly.

- Verifying BGP Neighbors on page 263
- Verifying BGP Groups on page 264
- Verifying BGP Summary Information on page 264
- Verifying That BGP Routes Are Installed in the Routing Table on page 265

Verifying BGP Neighbors

Purpose Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

Action From operational mode, enter the **show bgp neighbor** command.

```
user@A> show bgp neighbor
Peer: 192.163.6.4+179 AS 17    Local: 192.168.6.5+58852 AS 17
  Type: Internal    State: Established    Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-direct ]
  Options: Preference LocalAddress Refresh
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.163.6.4     Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            3
    Accepted prefixes:            3
    Suppressed due to damping:    0
    Advertised prefixes:          2
  Last traffic (seconds): Received 25   Sent 19   Checked 67
  Input messages:  Total 2420   Updates 4       Refreshes 0     Octets 46055
  Output messages: Total 2411   Updates 2       Refreshes 0     Octets 45921
  Output Queue[0]: 0

Peer: 192.168.40.4+179 AS 17   Local: 192.168.6.5+56466 AS 17
  Type: Internal    State: Established    Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
```

```
  Export: [ send-direct ]
  Options: Preference LocalAddress Refresh
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.40.4    Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            2
    Accepted prefixes:            2
    Suppressed due to damping:    0
    Advertised prefixes:          2
  Last traffic (seconds): Received 7    Sent 21   Checked 24
  Input messages:  Total 2412   Updates 2       Refreshes 0     Octets 45867
  Output messages: Total 2409   Updates 2       Refreshes 0     Octets 45883
  Output Queue[0]: 0
```

Verifying BGP Groups

Purpose Verify that the BGP groups are configured correctly.

Action From operational mode, enter the **show bgp group** command.

```
user@A> show bgp group
Group Type: Internal    AS: 17                      Local AS: 17
  Name: internal-peers  Index: 0                    Flags: <Export Eval>
  Export: [ send-direct ]
  Holdtime: 0
  Total peers: 2        Established: 2
  192.163.6.4+179
  192.168.40.4+179
  inet.0: 0/5/5/0

Groups: 1  Peers: 2    External: 0    Internal: 2    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 5          0          0          0          0          0
```

Verifying BGP Summary Information

Purpose Verify that the BGP configuration is correct.

Action From operational mode, enter the **show bgp summary** command.

```
user@A> show bgp summary
```

 Copyright © 2012, Juniper Networks, Inc.

```
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 5          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.163.6.4              17       2441       2432       0       0    18:18:52
0/3/3/0              0/0/0/0
192.168.40.4             17       2432       2430       0       0    18:18:48
0/2/2/0              0/0/0/0
```

Verifying That BGP Routes Are Installed in the Routing Table

Purpose Verify that the export policy configuration is causing the BGP routes to be installed in the routing tables of the peers.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@A> show route protocol bgp
inet.0: 7 destinations, 12 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30       [BGP/170] 07:09:57, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via ge-0/1/0.1
10.10.10.4/30       [BGP/170] 07:09:57, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via ge-0/1/0.1
                    [BGP/170] 07:07:12, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via ge-0/1/0.1
192.163.6.4/32      [BGP/170] 07:09:57, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via ge-0/1/0.1
192.168.40.4/32     [BGP/170] 07:07:12, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via ge-0/1/0.1
```

Related Documentation

- *Junos OS Policy Framework Configuration Guide*
- Understanding Internal BGP Peering Sessions
- BGP Configuration Overview on page 232

BGP Path Selections

- Understanding BGP Path Selection on page 265
- Example: Advertising Multiple Paths in BGP on page 269
- Understanding the Advertisement of Multiple Paths to a Single Destination in BGP on page 292
- Example: Ignoring the AS Path Attribute When Selecting the Best Path on page 293

Understanding BGP Path Selection

For each prefix in the routing table, the routing protocol process selects a single best path. After the best path is selected, the route is installed in the routing table. The best

path becomes the active route if the same prefix is not learned by a protocol with a lower (more preferred) global preference value, also known as the administrative distance. The algorithm for determining the active route is as follows:

1. Verify that the next hop can be resolved.
2. Choose the path with the lowest preference value (routing protocol process preference).

   Routes that are not eligible to be used for forwarding (for example, because they were rejected by routing policy or because a next hop is inaccessible) have a preference of –1 and are never chosen.
3. Prefer the path with higher local preference.

   For non-BGP paths, choose the path with the lowest **preference2** value.
4. If the accumulated interior gateway protocol (AIGP) attribute is enabled, prefer the path with the lower AIGP attribute.
5. Prefer the path with the shortest autonomous system (AS) path value (skipped if the **as-path-ignore** statement is configured).

   A confederation segment (sequence or set) has a path length of 0. An AS set has a path length of 1.
6. Prefer the route with the lower origin code.

   Routes learned from an IGP have a lower origin code than those learned from an exterior gateway protocol (EGP), and both have lower origin codes than incomplete routes (routes whose origin is unknown).
7. Prefer the path with the lowest multiple exit discriminator (MED) metric.

   Depending on whether nondeterministic routing table path selection behavior is configured, there are two possible cases:

   - If nondeterministic routing table path selection behavior is not configured (that is, if the **path-selection cisco-nondeterministic** statement is not included in the BGP configuration), for paths with the same neighboring AS numbers at the front of the AS path, prefer the path with the lowest MED metric. To always compare MEDs whether or not the peer ASs of the compared routes are the same, include the **path-selection always-compare-med** statement.
   - If nondeterministic routing table path selection behavior is configured (that is, the **path-selection cisco-nondeterministic** statement is included in the BGP configuration), prefer the path with the lowest MED metric.

   Confederations are not considered when determining neighboring ASs. A missing MED metric is treated as if a MED were present but zero.

NOTE: MED comparison works for single path selection within an AS (when the route does not include an AS path), though this usage Is uncommon.

Copyright © 2012, Juniper Networks, Inc.

8. Prefer strictly internal paths, which include IGP routes and locally generated routes (static, direct, local, and so forth).
9. Prefer strictly external BGP (EBGP) paths over external paths learned through internal BGP (IBGP) sessions.
10. Prefer the path whose next hop is resolved through the IGP route with the lowest metric.

    NOTE: A path is considered a BGP equal-cost path (and will be used for forwarding) if a tie-break is performed after the previous step. All paths with the same neighboring AS, learned by a multipath-enabled BGP neighbor, are considered.

    BGP multipath does not apply to paths that share the same MED-plus-IGP cost yet differ in IGP cost. Multipath path selection is based on the IGP cost metric, even if two paths have the same MED-plus-IGP cost.

11. If both paths are external, prefer the currently active path to minimize route-flapping. This rule is not used if:
    - **path-selection external-router-id** is configured.
    - Both peers have the same router ID.
    - Either peer is a confederation peer.
    - Neither path is the current active path.
12. Prefer the path from the peer with the lowest router ID. For any path with an originator ID attribute, substitute the originator ID for the router ID during router ID comparison.
13. Prefer the path with the shortest cluster list length. The length is 0 for no list.
14. Prefer the path from the peer with the lowest peer IP address.

By default, only the multiple exit discriminators (MEDs) of routes that have the same peer autonomous systems (ASs) are compared. You can configure routing table path selection options to obtain different behaviors.

The third step of the algorithm, by default, evaluates the length of the AS path and determines the active path. You can configure an option that enables Junos OS to skip this third step of the algorithm by including the **as-path-ignore** option.

NOTE: The **as-path-ignore** option is not supported for routing instances.

To configure routing table path selection behavior, include the **path-selection** statement:

```
path-selection {
   (always-compare-med | cisco-non-deterministic | external-router-id);
   as-path-ignore;
   med-plus-igp {
```

```
        igp-multiplier number;
        med-multiplier number;
    }
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Routing table path selection can be configured in one of the following ways:

- Using the same nondeterministic behavior as does the Cisco IOS software (**cisco-non-deterministic**). This behavior has two effects:
  - The active path is always first. All nonactive but eligible paths follow the active path and are maintained in the order in which they were received, with the most recent path first. Ineligible paths remain at the end of the list.
  - When a new path is added to the routing table, path comparisons are made without removing from consideration those paths that should never be selected because those paths lose the MED tie-breaking rule.

  NOTE: **The result of these two effects is that the system only sometimes compares the MED values between paths that it should otherwise compare. Because of this, we recommend that you not configure nondeterministic behavior.**

- Always comparing MEDs whether or not the peer ASs of the compared routes are the same (**always-compare-med**).
- Comparing the router ID between external BGP paths to determine the active path (**external-router-id**). By default, router ID comparison is not performed if one of the external paths is active. You can force the router ID comparison by restarting the routing process with the **restart routing** operational-mode command.
- Adding the IGP cost to the next-hop destination to the MED value before comparing MED values for path selection.

  BGP multipath does not apply to paths that share the same MED-plus-IGP cost, yet differ in IGP cost. Multipath path selection is based on the IGP cost metric, even if two paths have the same MED-plus-IGP cost.

Related Documentation

- Example: Ignoring the AS Path Attribute When Selecting the Best Path on page 293
- Example: Always Comparing MEDs on page 303

Copyright © 2012, Juniper Networks, Inc.

Example: Advertising Multiple Paths in BGP

In this example, BGP routers are configured to advertise multiple paths instead of advertising only the active path. Advertising multiple paths in BGP is specified in Internet draft draft-ietf-idr-add-paths-04.txt, *Advertisement of Multiple Paths in BGP*.

- Requirements on page 269
- Overview on page 269
- Configuration on page 270
- Verification on page 288

Requirements

This example uses the following hardware and software components:

- Eight BGP-speaking devices.
- Five of the BGP-enabled devices do not necessarily need to be routers. For example, they can be EX Series Ethernet Switches.
- Three of the BGP-enabled devices are configured to send multiple paths or receive multiple paths (or both send and receive multiple paths). These three BGP-enabled devices must be M Series Multiservice Edge Routers, MX Series 3D Universal Edge Routers, or T Series Core Routers.
- The three routers must be running Junos OS Release 11.4 or later.

Overview

In this example, Router R5, Router R6, and Router R7 redistribute static routes into BGP. Router R1 and Router R4 are route reflectors. Router R2 and Router R3 are clients to Route Reflector R1. Router R8 is a client to Route Reflector R4.

Route reflection is optional when multiple-path advertisement is enabled in BGP.

With the **add-path send path-count 6** configuration, Router R1 is configured to send up to six paths (per destination) to Router R4.

With the **add-path receive** configuration, Router R4 is configured to receive multiple paths from Router R1.

With the **add-path send path-count 6** configuration, Router R4 is also configured to send up to six paths to Router R8.

With the **add-path receive** configuration, Router R8 is configured to receive multiple paths from Router R4.

The **add-path send prefix-policy allow_199** policy configuration (along with the corresponding route filter) limits Router R4 to sending multiple paths for only the 199.1.1.1/32 route.

Copyright © 2012, Juniper Networks, Inc.

Topology Diagram

Figure 41 on page 270 shows the topology used in this example.

Figure 41: Advertisement of Multiple Paths in BGP

Configuration

- Configuring Router R1 on page 273
- Configuring Router R2 on page 275
- Configuring Router R3 on page 277
- Configuring Router R4 on page 279
- Configuring Router R5 on page 282
- Configuring Router R6 on page 283
- Configuring Router R7 on page 285
- Configuring Router R8 on page 286

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Router R1

```
set interfaces fe-0/0/0 unit 12 family inet address 10.0.12.1/24
set interfaces fe-0/0/1 unit 13 family inet address 10.0.13.1/24
set interfaces fe-1/0/0 unit 14 family inet address 10.0.14.1/24
set interfaces fe-1/2/0 unit 15 family inet address 10.0.15.1/24
set interfaces lo0 unit 10 family inet address 10.0.0.10/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.10
set protocols bgp group rr cluster 10.0.0.10
set protocols bgp group rr neighbor 10.0.0.20
set protocols bgp group rr neighbor 10.0.0.30
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.15.2 local-address 10.0.15.1
set protocols bgp group e1 neighbor 10.0.15.2 peer-as 2
set protocols bgp group rr_rr type internal
set protocols bgp group rr_rr local-address 10.0.0.10
```


Copyright © 2012, Juniper Networks, Inc.

```
set protocols bgp group rr_rr neighbor 10.0.0.40 family inet unicast add-path send
  path-count 6
set protocols ospf area 0.0.0.0 interface lo0.10 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.12
set protocols ospf area 0.0.0.0 interface fe-0/0/1.13
set protocols ospf area 0.0.0.0 interface fe-1/0/0.14
set protocols ospf area 0.0.0.0 interface fe-1/2/0.15
set routing-options router-id 10.0.0.10
set routing-options autonomous-system 1
```

Router R2

```
set interfaces fe-1/2/0 unit 21 family inet address 10.0.12.2/24
set interfaces fe-1/2/1 unit 26 family inet address 10.0.26.1/24
set interfaces lo0 unit 20 family inet address 10.0.0.20/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.20
set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.26.2 peer-as 2
set protocols ospf area 0.0.0.0 interface lo0.20 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.21
set protocols ospf area 0.0.0.0 interface fe-1/2/1.28
set policy-options policy-statement set_nh_self then next-hop self
set routing-options autonomous-system 1
```

Router R3

```
set interfaces fe-1/0/1 unit 31 family inet address 10.0.13.2/24
set interfaces fe-1/0/2 unit 37 family inet address 10.0.37.1/24
set interfaces lo0 unit 30 family inet address 10.0.0.30/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.30
set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.37.2 peer-as 2
set protocols ospf area 0.0.0.0 interface lo0.30 passive
set protocols ospf area 0.0.0.0 interface fe-1/0/1.31
set protocols ospf area 0.0.0.0 interface fe-1/0/2.37
set policy-options policy-statement set_nh_self then next-hop self
set routing-options autonomous-system 1
```

Router R4

```
set interfaces fe-1/2/0 unit 41 family inet address 10.0.14.2/24
set interfaces fe-1/2/1 unit 48 family inet address 10.0.48.1/24
set interfaces lo0 unit 40 family inet address 10.0.0.40/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.40
set protocols bgp group rr family inet unicast add-path receive
set protocols bgp group rr neighbor 10.0.0.10
set protocols bgp group rr_client type internal
set protocols bgp group rr_client local-address 10.0.0.40
set protocols bgp group rr_client cluster 10.0.0.40
set protocols bgp group rr_client neighbor 10.0.0.80 family inet unicast add-path send
  path-count 6
set protocols bgp group rr_client neighbor 10.0.0.80 family inet unicast add-path send
  prefix-policy allow_199
set protocols ospf area 0.0.0.0 interface fe-1/2/0.41
set protocols ospf area 0.0.0.0 interface lo0.40 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/1.48
```

```
set routing-options autonomous-system 1
set policy-options policy-statement allow_199 from route-filter 199.1.1.1/32 exact
set policy-options policy-statement allow_199 then accept
```

Router R5

```
set interfaces fe-1/2/0 unit 51 family inet address 10.0.15.2/24
set interfaces lo0 unit 50 family inet address 10.0.0.50/32
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.15.1 export s2b
set protocols bgp group e1 neighbor 10.0.15.1 peer-as 1
set policy-options policy-statement s2b from protocol static
set policy-options policy-statement s2b from protocol direct
set policy-options policy-statement s2b then as-path-expand 2
set policy-options policy-statement s2b then accept
set routing-options autonomous-system 2
set routing-options static route 199.1.1.1/32 reject
set routing-options static route 198.1.1.1/32 reject
```

Router R6

```
set interfaces fe-1/2/0 unit 62 family inet address 10.0.26.2/24
set interfaces lo0 unit 60 family inet address 10.0.0.60/32
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.26.1 export s2b
set protocols bgp group e1 neighbor 10.0.26.1 peer-as 1
set policy-options policy-statement s2b from protocol static
set policy-options policy-statement s2b from protocol direct
set policy-options policy-statement s2b then accept
set routing-options autonomous-system 2
set routing-options static route 199.1.1.1/32 reject
set routing-options static route 198.1.1.1/32 reject
```

Router R7

```
set interfaces fe-1/2/0 unit 73 family inet address 10.0.37.2/24
set interfaces lo0 unit 70 family inet address 10.0.0.70/32
set policy-options policy-statement s2b from protocol static
set policy-options policy-statement s2b from protocol direct
set policy-options policy-statement s2b then accept
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.37.1 export s2b
set protocols bgp group e1 neighbor 10.0.37.1 peer-as 1
set routing-options autonomous-system 2
set routing-options static route 199.1.1.1/32 reject
```

Router R8

```
set interfaces fe-1/2/0 unit 84 family inet address 10.0.48.2/24
set interfaces lo0 unit 80 family inet address 10.0.0.80/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.80
set protocols bgp group rr neighbor 10.0.0.40 family inet unicast add-path receive
set protocols ospf area 0.0.0.0 interface lo0.80 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.84
set routing-options autonomous-system 1
```


Copyright © 2012, Juniper Networks, Inc.

Configuring Router R1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Router R1:

1. Configure the interfaces to Router R2, Router R3, Router R5, and Router R4, and configure the loopback (**lo0**) interface.

```
[edit interfaces]
user@R1# set fe-0/0/0 unit 12 family inet address 10.0.12.1/24

user@R1# set fe-0/0/1 unit 13 family inet address 10.0.13.1/24

user@R1# set fe-1/0/0 unit 14 family inet address 10.0.14.1/24

user@R1# set fe-1/2/0 unit 15 family inet address 10.0.15.1/24

user@R1#set lo0 unit 10 family inet address 10.0.0.10/32
```

2. Configure BGP on the interfaces, and configure IBGP route reflection.

```
[edit protocols bgp]
user@R1# set group rr type internal
user@R1# set group rr local-address 10.0.0.10
user@R1# set group rr cluster 10.0.0.10
user@R1# set group rr neighbor 10.0.0.20
user@R1# set group rr neighbor 10.0.0.30

user@R1# set group rr_rr type internal
user@R1# set group rr_rr local-address 10.0.0.10

user@R1# set group e1 type external
user@R1# set group e1 neighbor 10.0.15.2 local-address 10.0.15.1
user@R1# set group e1 neighbor 10.0.15.2 peer-as 2
```

3. Configure Router R1 to send up to six paths to its neighbor, Router R4.

   The destination of the paths can be any destination that Router R1 can reach through multiple paths.

```
[edit protocols bgp]
user@R1# set group rr_rr neighbor 10.0.0.40 family inet unicast add-path send
  path-count 6
```

4. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@R1# set area 0.0.0.0 interface lo0.10 passive
user@R1# set area 0.0.0.0 interface fe-0/0/0.12
user@R1# set area 0.0.0.0 interface fe-0/0/1.13
user@R1# set area 0.0.0.0 interface fe-1/0/0.14
user@R1# set area 0.0.0.0 interface fe-1/2/0.15
```

Copyright © 2012, Juniper Networks, Inc.

5. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@R1# set router-id 10.0.0.10
user@R1# set autonomous-system 1
```

6. If you are done configuring the device, commit the configuration.

```
user@R1# commit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show interfaces
fe-0/0/0 {
   unit 12 {
      family inet {
         address 10.0.12.1/24;
      }
   }
}
fe-0/0/1 {
   unit 13 {
      family inet {
         address 10.0.13.1/24;
      }
   }
}
fe-1/0/0 {
   unit 14 {
      family inet {
         address 10.0.14.1/24;
      }
   }
}
fe-1/2/0 {
   unit 15 {
      family inet {
         address 10.0.15.1/24;
      }
   }
}
lo0 {
   unit 10 {
      family inet {
         address 10.0.0.10/32;
      }
   }
}

user@R1# show protocols
bgp {
   group rr {
      type internal;
      local-address 10.0.0.10;
```

 Copyright © 2012, Juniper Networks, Inc.

```
        cluster 10.0.0.10;
        neighbor 10.0.0.20;
        neighbor 10.0.0.30;
      }
      group e1 {
        type external;
        neighbor 10.0.15.2 {
          local-address 10.0.15.1;
          peer-as 2;
        }
      }
      group rr_rr {
        type internal;
        local-address 10.0.0.10;
        neighbor 10.0.0.40 {
          family inet {
            unicast {
              add-path {
                send {
                  path-count 6;
                }
              }
            }
          }
        }
      }
    }
    ospf {
      area 0.0.0.0 {
        interface lo0.10 {
          passive;
        }
        interface fe-0/0/0.12;
        interface fe-0/0/1.13;
        interface fe-1/0/0.14;
        interface fe-1/2/0.15;
      }
    }

    user@R1# show routing-options
    router-id 10.0.0.10;
    autonomous-system 1;
```

Configuring Router R2

Step-by-Step Procedure

To configure Router R2:

1. Configure the loopback (**lo0**) interface and the interfaces to Router R6 and Router R1.

```
[edit interfaces]
user@R2# set fe-1/2/0 unit 21 family inet address 10.0.12.2/24

user@R2# set fe-1/2/1 unit 26 family inet address 10.0.26.1/24

user@R2# set lo0 unit 20 family inet address 10.0.0.20/32
```

2. Configure BGP and OSPF on Router R2's interfaces.

```
[edit protocols]
user@R2# set bgp group rr type internal
user@R2# set bgp group rr local-address 10.0.0.20

user@R2# set bgp group e1 type external
user@R2# set bgp group e1 neighbor 10.0.26.2 peer-as 2

user@R2# set ospf area 0.0.0.0 interface lo0.20 passive
user@R2# set ospf area 0.0.0.0 interface fe-1/2/0.21
user@R2# set ospf area 0.0.0.0 interface fe-1/2/1.28
```

3. For routes sent from Router R2 to Router R1, advertise Router R2 as the next hop, because Router R1 does not have a route to Router R6's address on the 10.0.26.0/24 network.

```
[edit]
user@R2# set policy-options policy-statement set_nh_self then next-hop self
user@R2# set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
```

4. Configure the autonomous system number.

```
[edit]
user@R2# set routing-options autonomous-system 1
```

5. If you are done configuring the device, commit the configuration.

```
user@R2# commit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R2# show interfaces
fe-1/2/0 {
   unit 21 {
      family inet {
         address 10.0.12.2/24;
      }
   }
}
fe-1/2/1 {
   unit 26 {
      family inet {
         address 10.0.26.1/24;
      }
   }
}
lo0 {
   unit 20 {
      family inet {
         address 10.0.0.20/32;
      }
   }
```

 Copyright © 2012, Juniper Networks, Inc.

```
}

user@R2# show policy-options
policy-statement set_nh_self {
   then {
      next-hop self;
   }
}

user@R2# show protocols
bgp {
   group rr {
      type internal;
      local-address 10.0.0.20;
      neighbor 10.0.0.10 {
         export set_nh_self;
      }
   }
   group e1 {
      type external;
      neighbor 10.0.26.2 {
         peer-as 2;
      }
   }
}
ospf {
   area 0.0.0.0 {
      interface lo0.20 {
         passive;
      }
      interface fe-1/2/0.21;
      interface fe-1/2/1.28;
   }
}

user@R2# show routing-options
autonomous-system 1;
```

Configuring Router R3

Step-by-Step Procedure

To configure Router R3:

1. Configure the loopback (**lo0**) interface and the interfaces to Router R7 and Router R1.

```
[edit interfaces]
user@R3# set fe-1/0/1 unit 31 family inet address 10.0.13.2/24

user@R3# set fe-1/0/2 unit 37 family inet address 10.0.37.1/24

user@R3# set lo0 unit 30 family inet address 10.0.0.30/32
```

2. Configure BGP and OSPF on Router R3's interfaces.

```
[edit protocols]
user@R3# set bgp group rr type internal
user@R3# set bgp group rr local-address 10.0.0.30
```

```
user@R3# set bgp group e1 type external
user@R3# set bgp group e1 neighbor 10.0.37.2 peer-as 2

user@R3# set ospf area 0.0.0.0 interface lo0.30 passive
user@R3# set ospf area 0.0.0.0 interface fe-1/0/1.31
user@R3# set ospf area 0.0.0.0 interface fe-1/0/2.37
```

3. For routes sent from Router R3 to Router R1, advertise Router R3 as the next hop, because Router R1 does not have a route to Router R7's address on the 10.0.37.0/24 network.

```
[edit]
user@R3# set policy-options policy-statement set_nh_self then next-hop self
user@R3# set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
```

4. Configure the autonomous system number.

```
[edit]
user@R3# set routing-options autonomous-system 1
```

5. If you are done configuring the device, commit the configuration.

```
user@R3# commit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R3# show interfaces
fe-1/0/1 {
   unit 31 {
      family inet {
         address 10.0.13.2/24;
      }
   }
}
fe-1/0/2 {
   unit 37 {
      family inet {
         address 10.0.37.1/24;
      }
   }
}
lo0 {
   unit 30 {
      family inet {
         address 10.0.0.30/32;
      }
   }
}

user@R3# show policy-options
policy-statement set_nh_self {
   then {
      next-hop self;
   }
```

 Copyright © 2012, Juniper Networks, Inc.

```
}

user@R3# show protocols
bgp {
  group rr {
    type internal;
    local-address 10.0.0.30;
    neighbor 10.0.0.10 {
      export set_nh_self;
    }
  }
  group e1 {
    type external;
    neighbor 10.0.37.2 {
      peer-as 2;
    }
  }
}
  ospf {
    area 0.0.0.0 {
      interface lo0.30 {
        passive;
      }
      interface fe-1/0/1.31;
      interface fe-1/0/2.37;
    }
  }

user@R3# show routing-options
autonomous-system 1;
```

Configuring Router R4

Step-by-Step Procedure

To configure Router R4:

1. Configure the interfaces to Router R1 and Router R8, and configure the loopback (**lo0**) interface.

```
[edit interfaces]
user@R4# set fe-1/2/0 unit 41 family inet address 10.0.14.2/24

user@R4# set fe-1/2/1 unit 48 family inet address 10.0.48.1/24

user@R4# set lo0 unit 40 family inet address 10.0.0.40/32
```

2. Configure BGP on the interfaces, and configure IBGP route reflection.

```
[edit protocols bgp]
user@R4# set group rr type internal
user@R4# set group rr local-address 10.0.0.40
user@R4# set group rr neighbor 10.0.0.10

user@R4# set group rr_client type internal
user@R4# set group rr_client local-address 10.0.0.40
user@R4# set group rr_client cluster 10.0.0.40
```

3. Configure Router R4 to send up to six paths to its neighbor, Router R8.

   The destination of the paths can be any destination that Router R4 can reach through multiple paths.

   ```
   [edit protocols bgp]
   user@R4# set group rr_client neighbor 10.0.0.80 family inet unicast add-path send
     path-count 6
   ```

4. Configure Router R4 to receive multiple paths from its neighbor, Router R1.

   The destination of the paths can be any destination that Router R1 can reach through multiple paths.

   ```
   [edit protocols bgp]
   user@R4# set group rr family inet unicast add-path receive
   ```

5. Configure OSPF on the interfaces.

   ```
   [edit protocols ospf]
   user@R4# set area 0.0.0.0 interface fe-1/2/0.41
   user@R4# set area 0.0.0.0 interface lo0.40 passive
   user@R4# set area 0.0.0.0 interface fe-1/2/1.48
   ```

6. Configure a policy that allows Router R4 to send Router R8 multiple paths to the 199.1.1.1/32 route.

   Router R4 receives multiple paths for the 198.1.1.1/32 route and the 199.1.1.1/32 route. However, because of this policy, Router R4 only sends multiple paths for the 199.1.1.1/32 route.

   ```
   [edit]
   user@R4# set protocols bgp group rr_client neighbor 10.0.0.80 family inet unicast
     add-path send prefix-policy allow_199
   user@R4# set policy-options policy-statement allow_199 from route-filter 199.1.1.1/32
     exact
   user@R4# set policy-options policy-statement allow_199 then accept
   ```

7. Configure the autonomous system number.

   ```
   [edit routing-options]
   user@R4# set autonomous-system 1
   ```

8. If you are done configuring the device, commit the configuration.

   ```
   user@R4# commit
   ```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R4# show interfaces
fe-1/2/0 {
   unit 41 {
      family inet {
         address 10.0.14.2/24;
      }
```

 Copyright © 2012, Juniper Networks, Inc.

```
    }
  }
  fe-1/2/1 {
    unit 48 {
      family inet {
        address 10.0.48.1/24;
      }
    }
  }
  lo0 {
    unit 40 {
      family inet {
        address 10.0.0.40/32;
      }
    }
  }
```

```
user@R4# show policy-options
policy-statement allow_199 {
  from {
    route-filter 199.1.1.1/32 exact;
  }
  then accept;
}
```

```
user@R4# show protocols
bgp {
  group rr {
    type internal;
    local-address 10.0.0.40;
    family inet {
      unicast {
        add-path {
          receive;
        }
      }
    }
    neighbor 10.0.0.10;
  }
  group rr_client {
    type internal;
    local-address 10.0.0.40;
    cluster 10.0.0.40;
    neighbor 10.0.0.80 {
      family inet {
        unicast {
          add-path {
            send {
              path-count 6;
              prefix-policy allow_199;
            }
          }
        }
      }
    }
  }
}
```

```
ospf {
  area 0.0.0.0 {
    interface lo0.40 {
      passive;
    }
    interface fe-1/2/0.41;
    interface fe-1/2/1.48;
  }
}
```

```
user@R4# show routing-options
autonomous-system 1;
```

Configuring Router R5

Step-by-Step Procedure

To configure Router R5:

1. Configure the loopback (**lo0**) interface and the interface to Router R1.

```
[edit interfaces]
user@R5# set fe-1/2/0 unit 51 family inet address 10.0.15.2/24

user@R5# set lo0 unit 50 family inet address 10.0.0.50/32
```

2. Configure BGP on Router R5's interface.

```
[edit protocols]
user@R5# set bgp group e1 type external
user@R5# set bgp group e1 neighbor 10.0.15.1 peer-as 1
```

3. Create static routes for redistribution into BGP.

```
[edit]
user@R5# set routing-options static route 199.1.1.1/32 reject
user@R5# set routing-options static route 198.1.1.1/32 reject
```

4. Redistribute static and direct routes into BGP.

```
[edit]
user@R5# set protocols bgp group e1 neighbor 10.0.15.1 export s2b
user@R5# set policy-options policy-statement s2b from protocol static
user@R5# set policy-options policy-statement s2b from protocol direct
user@R5# set policy-options policy-statement s2b then as-path-expand 2
user@R5# set policy-options policy-statement s2b then accept
```

5. Configure the autonomous system number.

```
[edit]
user@R5# set routing-options autonomous-system 2
```

6. If you are done configuring the device, commit the configuration.

```
user@R5# commit
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Copyright © 2012, Juniper Networks, Inc.

```
user@R5# show interfaces
fe-1/2/0 {
   unit 51 {
      family inet {
         address 10.0.15.2/24;
      }
   }
}
lo0 {
   unit 50 {
      family inet {
         address 10.0.0.50/32;
      }
   }
}
```

```
user@R5# show policy-options
policy-statement s2b {
   from protocol [ static direct ];
   then {
      as-path-expand 2;
      accept;
   }
}
```

```
user@R5# show protocols
bgp {
   group e1 {
      type external;
      neighbor 10.0.15.1 {
         export s2b;
         peer-as 1;
      }
   }
}
```

```
user@R5# show routing-options
static {
   route 198.1.1.1/32 reject;
   route 199.1.1.1/32 reject;
}
autonomous-system 2;
```

Configuring Router R6

Step-by-Step Procedure

To configure Router R6:

1. Configure the loopback (**lo0**) interface and the interface to Router R2.

```
[edit interfaces]
user@R6# set fe-1/2/0 unit 62 family inet address 10.0.26.2/24

user@R6# set lo0 unit 60 family inet address 10.0.0.60/32
```

2. Configure BGP on Router R6's interface.

```
[edit protocols]
user@R6# set bgp group e1 type external
```

```
user@R6# set bgp group e1 neighbor 10.0.26.1 peer-as 1
```

3. Create static routes for redistribution into BGP.

```
[edit]
user@R6# set routing-options static route 199.1.1.1/32 reject
user@R6# set routing-options static route 198.1.1.1/32 reject
```

4. Redistribute static and direct routes from Router R6's routing table into BGP.

```
[edit]
user@R6# set protocols bgp group e1 neighbor 10.0.26.1 export s2b
user@R6# set policy-options policy-statement s2b from protocol static
user@R6# set policy-options policy-statement s2b from protocol direct
user@R6# set policy-options policy-statement s2b then accept
```

5. Configure the autonomous system number.

```
[edit]
user@R6# set routing-options autonomous-system 2
```

6. If you are done configuring the device, commit the configuration.

```
user@R6# commit
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R6# show interfaces
fe-1/2/0 {
   unit 62 {
      family inet {
         address 10.0.26.2/24;
      }
   }
}
lo0 {
   unit 60 {
      family inet {
         address 10.0.0.60/32;
      }
   }
}
```

```
user@R6# show policy-options
policy-statement s2b {
   from protocol [ static direct ];
   then accept;
}
```

```
user@R6# show protocols
bgp {
   group e1 {
      type external;
      neighbor 10.0.26.1 {
         export s2b;
```

 Copyright © 2012, Juniper Networks, Inc.

```
            peer-as 1;
        }
    }
}

user@R6# show routing-options
static {
    route 198.1.1.1/32 reject;
    route 199.1.1.1/32 reject;
}
autonomous-system 2;
```

Configuring Router R7

Step-by-Step Procedure

To configure Router R7:

1. Configure the loopback (**lo0**) interface and the interface to Router R3.

```
[edit interfaces]
user@R7# set fe-1/2/0 unit 73 family inet address 10.0.37.2/24

user@R7# set lo0 unit 70 family inet address 10.0.0.70/32
```

2. Configure BGP on Router R7's interface.

```
[edit protocols]
user@R7# set bgp group e1 type external
user@R7# set bgp group e1 neighbor 10.0.37.1 peer-as 1
```

3. Create a static route for redistribution into BGP.

```
[edit]
user@R7# set routing-options static route 199.1.1.1/32 reject
```

4. Redistribute static and direct routes from Router R7's routing table into BGP.

```
[edit]
user@R7# set protocols bgp group e1 neighbor 10.0.37.1 export s2b
user@R7# set policy-options policy-statement s2b from protocol static
user@R7# set policy-options policy-statement s2b from protocol direct
user@R7# set policy-options policy-statement s2b then accept
```

5. Configure the autonomous system number.

```
[edit]
user@R7# set routing-options autonomous-system 2
```

6. If you are done configuring the device, commit the configuration.

```
user@R7# commit
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R7# show interfaces
fe-1/2/0 {
```

```
    unit 73 {
      family inet {
        address 10.0.37.2/24;
      }
    }
  }
  lo0 {
    unit 70 {
      family inet {
        address 10.0.0.70/32;
      }
    }
  }

user@R7# show policy-options
policy-statement s2b {
  from protocol [ static direct ];
  then accept;
}

user@R7# show protocols
bgp {
  group e1 {
    type external;
    neighbor 10.0.37.1 {
      export s2b;
      peer-as 1;
    }
  }
}

user@R7# show routing-options
static {
  route 199.1.1.1/32 reject;
}
autonomous-system 2;
```

Configuring Router R8

Step-by-Step Procedure

To configure Router R8:

1. Configure the loopback (**lo0**) interface and the interface to Router R4.

```
[edit interfaces]
user@R8# set fe-1/2/0 unit 84 family inet address 10.0.48.2/24

user@R8# set lo0 unit 80 family inet address 10.0.0.80/32
```

2. Configure BGP and OSPF on Router R8's interface.

```
[edit protocols]
user@R8# set bgp group rr type internal
user@R8# set bgp group rr local-address 10.0.0.80

user@R8# set ospf area 0.0.0.0 interface lo0.80 passive
user@R8# set ospf area 0.0.0.0 interface fe-1/2/0.84
```


Copyright © 2012, Juniper Networks, Inc.

3. Configure Router R8 to receive multiple paths from its neighbor, Router R4.

   The destination of the paths can be any destination that Router R4 can reach through multiple paths.

   ```
   [edit protocols]
   user@R8# set bgp group rr neighbor 10.0.0.40 family inet unicast add-path receive
   ```

4. Configure the autonomous system number.

   ```
   [edit]
   user@R8# set routing-options autonomous-system 1
   ```

5. If you are done configuring the device, commit the configuration.

   ```
   user@R8# commit
   ```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R8# show interfaces
fe-1/2/0 {
   unit 84 {
      family inet {
         address 10.0.48.2/24;
      }
   }
}
lo0 {
   unit 80 {
      family inet {
         address 10.0.0.80/32;
      }
   }
}

user@R8# show protocols
bgp {
   group rr {
      type internal;
      local-address 10.0.0.80;
      neighbor 10.0.0.40 {
         family inet {
            unicast {
               add-path {
                  receive;
               }
            }
         }
      }
   }
}
ospf {
   area 0.0.0.0 {
      interface lo0.80 {
         passive;
```

```
        }
        interface fe-1/2/0.84;
      }
    }
```

```
user@R8# show routing-options
autonomous-system 1;
```

Verification

- Verifying That the BGP Peers Have the Ability to Send and Receive Multiple Paths on page 288
- Verifying That Router R1 Is Advertising Multiple Paths on page 289
- Verifying That Router R4 Is Receiving and Advertising Multiple Paths on page 289
- Verifying That Router R8 Is Receiving Multiple Paths on page 290
- Checking the Path ID on page 290

Verifying That the BGP Peers Have the Ability to Send and Receive Multiple Paths

Purpose Make sure that one or both of the following strings appear in the output of the **show bgp neighbor** command:

- **NLRI's for which peer can receive multiple paths: inet-unicast**
- **NLRI's for which peer can send multiple paths: inet-unicast**

Action

```
user@R1> show bgp neighbor 10.0.0.40
Peer: 10.0.0.40+179 AS 1       Local: 10.0.0.10+65237 AS 1
  Type: Internal    State: Established    Flags: <Sync>
...  NLRI's for which peer can receive multiple paths: inet-unicast
...

user@R4> show bgp neighbor 10.0.0.10
Peer: 10.0.0.10+65237 AS 1     Local: 10.0.0.40+179 AS 1
  Type: Internal    State: Established    Flags: <Sync>
...
  NLRI's for which peer can send multiple paths: inet-unicast
...

user@R4> show bgp neighbor 10.0.0.80
Peer: 10.0.0.80+55416 AS 1     Local: 10.0.0.40+179 AS 1
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  ,,,
  NLRI's for which peer can receive multiple paths: inet-unicast
  ...

user@R8> show bgp neighbor 10.0.0.40
Peer: 10.0.0.40+179 AS 1       Local: 10.0.0.80+55416 AS 1
  Type: Internal    State: Established    Flags: <Sync>
  ...
  NLRI's for which peer can send multiple paths: inet-unicast
  ...
```

 Copyright © 2012, Juniper Networks, Inc.

Verifying That Router R1 Is Advertising Multiple Paths

Purpose Make sure that multiple paths to the 198.1.1.1/32 destination and multiple paths to the 199.1.1.1/32 destination are advertised to Router R4.

Action

```
user@R1> show route advertising-protocol bgp 10.0.0.40
inet.0: 21 destinations, 25 routes (21 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.50/32            10.0.15.2                    100        2 2 I
* 10.0.0.60/32            10.0.0.20                    100        2 I
* 10.0.0.70/32            10.0.0.30                    100        2 I
* 198.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.15.2                    100        2 2 I
* 199.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.0.30                    100        2 I
                          10.0.15.2                    100        2 2 I
* 200.1.1.0/30            10.0.0.20                    100        2 I
```

Meaning When you see one prefix and more than one next hop, it means that multiple paths are advertised to Router R4.

Verifying That Router R4 Is Receiving and Advertising Multiple Paths

Purpose Make sure that multiple paths to the 199.1.1.1/32 destination are received from Router R1 and advertised to Router R8. Make sure that multiple paths to the 198.1.1.1/32 destination are received from Router R1, but only one path to this destination is advertised to Router R8.

Action

```
user@R4> show route receive-protocol bgp 10.0.0.10
inet.0: 19 destinations, 22 routes (19 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.50/32            10.0.15.2                    100        2 2 I
* 10.0.0.60/32            10.0.0.20                    100        2 I
* 10.0.0.70/32            10.0.0.30                    100        2 I
* 198.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.15.2                    100        2 2 I
* 199.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.0.30                    100        2 I
                          10.0.15.2                    100        2 2 I
* 200.1.1.0/30            10.0.0.20                    100        2 I
```

```
user@R4> show route advertising-protocol bgp 10.0.0.80
inet.0: 19 destinations, 22 routes (19 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.50/32            10.0.15.2                    100        2 2 I
* 10.0.0.60/32            10.0.0.20                    100        2 I
* 10.0.0.70/32            10.0.0.30                    100        2 I
* 198.1.1.1/32            10.0.0.20                    100        2 I
* 199.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.0.30                    100        2 I
                          10.0.15.2                    100        2 2 I
* 200.1.1.0/30            10.0.0.20                    100        2 I
```

Copyright © 2012, Juniper Networks, Inc.

Meaning The **show route receive-protocol** command shows that Router R4 receives two paths to the 198.1.1.1/32 destination and three paths to the 199.1.1.1/32 destination. The **show route advertising-protocol** command shows that Router R4 advertises only one path to the 198.1.1.1/32 destination and advertises all three paths to the 199.1.1.1/32 destination.

Because of the prefix-policy that is applied to Router R4, Router R4 does not advertise multiple paths to the 198.1.1.1/32 destination. Router R4 advertises only one path to the 198.1.1.1/32 destination even though it receives multiple paths to this destination.

Verifying That Router R8 Is Receiving Multiple Paths

Purpose Make sure that Router R8 receives multiple paths to the 199.1.1.1/32 destination through Router R4. Make sure that Router R8 receives only one path to the 198.1.1.1/32 destination through Router R4.

Action

```
user@R8> show route receive-protocol bgp 10.0.0.40
inet.0: 18 destinations, 20 routes (18 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.50/32            10.0.15.2                    100        2 2 I
* 10.0.0.60/32            10.0.0.20                    100        2 I
* 10.0.0.70/32            10.0.0.30                    100        2 I
* 198.1.1.1/32            10.0.0.20                    100        2 I
* 199.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.0.30                    100        2 I
                          10.0.15.2                    100        2 2 I
* 200.1.1.0/30            10.0.0.20                    100        2 I
```

Checking the Path ID

Purpose On the downstream devices, Router R4 and Router R8, verify that a path ID uniquely identifies the path. Look for the **Addpath Path ID:** string.

Action

```
user@R4> show route 199.1.1.1/32 detail

inet.0: 18 destinations, 20 routes (18 active, 0 holddown, 0 hidden)
199.1.1.1/32 (3 entries, 3 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 9
                Source: 10.0.0.10
                Next hop type: Router, Next hop index: 676
                Next hop: 10.0.14.1 via lt-1/2/0.41, selected
                Protocol next hop: 10.0.0.20
                Indirect next hop: 92041c8 262146
                State: <Active Int Ext>
                Local AS:     1 Peer AS:     1
                Age: 1:44:37    Metric2: 2
                Task: BGP_1.10.0.0.10+65237
                Announcement bits (3): 2-KRT 3-BGP RT Background 4-Resolve tree
1
                AS path: 2 I (Originator) Cluster list:  10.0.0.10
                AS path:  Originator ID: 10.0.0.20
                Accepted
                Localpref: 100
                Router ID: 10.0.0.10
                Addpath Path ID: 1
         BGP    Preference: 170/-101
```

 Copyright © 2012, Juniper Networks, Inc.

```
                    Next hop type: Indirect
                    Next-hop reference count: 4
                    Source: 10.0.0.10
                    Next hop type: Router, Next hop index: 676
                    Next hop: 10.0.14.1 via lt-1/2/0.41, selected
                    Protocol next hop: 10.0.0.30
                    Indirect next hop: 92042ac 262151
                    State: <NotBest Int Ext>
                    Inactive reason: Not Best in its group - Router ID
                    Local AS:     1 Peer AS:     1
                    Age: 1:44:37    Metric2: 2
                    Task: BGP_1.10.0.0.10+65237
                    Announcement bits (1): 3-BGP RT Background
                    AS path: 2 I (Originator) Cluster list:  10.0.0.10
                    AS path:  Originator ID: 10.0.0.30
                    Accepted
                    Localpref: 100
                    Router ID: 10.0.0.10
                    Addpath Path ID: 2
             BGP    Preference: 170/-101
                    Next hop type: Indirect
                    Next-hop reference count: 4
                    Source: 10.0.0.10
                    Next hop type: Router, Next hop index: 676
                    Next hop: 10.0.14.1 via lt-1/2/0.41, selected
                    Protocol next hop: 10.0.15.2
                    Indirect next hop: 92040e4 262150
                    State: <Int Ext>
                    Inactive reason: AS path
                    Local AS:     1 Peer AS:     1
                    Age: 1:44:37    Metric2: 2
                    Task: BGP_1.10.0.0.10+65237
                    Announcement bits (1): 3-BGP RT Background
                    AS path: 2 2 I
                    Accepted
                    Localpref: 100
                    Router ID: 10.0.0.10
                    Addpath Path ID: 3
```

```
user@R8> show route 199.1.1.1/32 detail

inet.0: 17 destinations, 19 routes (17 active, 0 holddown, 0 hidden)
199.1.1.1/32 (3 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 9
                Source: 10.0.0.40
                Next hop type: Router, Next hop index: 1045
                Next hop: 10.0.48.1 via lt-1/2/0.84, selected
                Protocol next hop: 10.0.0.20
                Indirect next hop: 91fc0e4 262148
                State: <Active Int Ext>
                Local AS:     1 Peer AS:     1
                Age: 1:56:51    Metric2: 3
                Task: BGP_1.10.0.0.40+179
                Announcement bits (2): 2-KRT 4-Resolve tree 1
                AS path: 2 I (Originator) Cluster list:  10.0.0.40 10.0.0.10
                AS path:  Originator ID: 10.0.0.20
                Accepted
                Localpref: 100
                Router ID: 10.0.0.40
```

```
        Addpath Path ID: 1
BGP     Preference: 170/-101
        Next hop type: Indirect
        Next-hop reference count: 4
        Source: 10.0.0.40
        Next hop type: Router, Next hop index: 1045
        Next hop: 10.0.48.1 via lt-1/2/0.84, selected
        Protocol next hop: 10.0.0.30
        Indirect next hop: 91fc1c8 262152
        State: <NotBest Int Ext>
        Inactive reason: Not Best in its group - Router ID
        Local AS:     1 Peer AS:     1
        Age: 1:56:51    Metric2: 3
        Task: BGP_1.10.0.0.40+179
        AS path: 2 I (Originator) Cluster list:  10.0.0.40 10.0.0.10
        AS path:  Originator ID: 10.0.0.30
        Accepted
        Localpref: 100
        Router ID: 10.0.0.40
        Addpath Path ID: 2
BGP     Preference: 170/-101
        Next hop type: Indirect
        Next-hop reference count: 4
        Source: 10.0.0.40
        Next hop type: Router, Next hop index: 1045
        Next hop: 10.0.48.1 via lt-1/2/0.84, selected
        Protocol next hop: 10.0.15.2
        Indirect next hop: 91fc2ac 262153
        State: <Int Ext>
        Inactive reason: AS path
        Local AS:     1 Peer AS:     1
        Age: 1:56:51    Metric2: 3
        Task: BGP_1.10.0.0.40+179
        AS path: 2 2 I (Originator) Cluster list:  10.0.0.40
        AS path:  Originator ID: 10.0.0.10
        Accepted
        Localpref: 100
        Router ID: 10.0.0.40
        Addpath Path ID: 3
```

Related Documentation

- Understanding the Advertisement of Multiple Paths to a Single Destination in BGP on page 292

Understanding the Advertisement of Multiple Paths to a Single Destination in BGP

BGP peers advertise routes to each other in update messages. BGP stores its routes in the Junos OS routing table (**inet.0**). For each prefix in the routing table, the routing protocol process selects a single best path, called the active path. Unless you configure BGP to advertise multiple paths to the same destination, BGP advertises only the active path.

Instead of advertising only the active path to a destination, you can configure BGP to advertise multiple paths to the destination. Within an autonomous system (AS), the availability of multiple exit points to reach a destination provides the following benefits:

- Fault tolerance—Path diversity leads to reduction in restoration time after failure. For instance, a border router after receiving multiple paths to the same destination can precompute a backup path and have it ready so that when the primary path becomes

 Copyright © 2012, Juniper Networks, Inc.

invalid, the border router can use the backup to quickly restore connectivity. Without a backup path, the restoration time depends on BGP reconvergence, which includes withdraw and advertisement messages in the network before a new best path can be learned.

- Load balancing—The availability of multiple paths to reach the same destination enables load balancing of traffic, if the routing within the AS meets certain constraints.
- Maintenance—The availability of alternate exit points allows for graceful maintenance operation of routers.

The following limitations apply to advertising multiple routes in BGP:

- IPv4 unicast (**family inet unicast**) routes only.
- Internal BGP (IBGP) peers only. No support on external BGP (EBGP) peers.
- Master instance only. No support for routing instances.
- No support for nonstop active routing (NSR).
- No BGP Monitoring Protocol (BMP) support.
- No support for EBGP sessions between confederations.
- Prefix policies enable you to filter routes on a router that is configured to advertise multiple paths to a destination. However, prefix policies can only match routes. Prefix policies cannot change the attributes of routes.

Related Documentation

- Understanding BGP Path Selection on page 265
- Example: Advertising Multiple Paths in BGP on page 269

Example: Ignoring the AS Path Attribute When Selecting the Best Path

If multiple BGP routes to the same destination exist, BGP selects the best path based on the route attributes of the paths. One of the route attributes that affects the best-path decision is the length of the AS paths of each route. Routes with shorter AS paths are preferred over those with longer AS paths. Although not typically practical, some scenarios might require that the AS path length be ignored in the route selection process. This example shows how to configure a routing device to ignore the AS path attribute.

- Requirements on page 293
- Overview on page 294
- Configuration on page 295
- Verification on page 300

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

On externally connected routing devices, the purpose of skipping the AS path comparison might be to force an external BGP (EBGP) versus internal BGP (IBGP) decision to remove traffic from your network as soon as possible. On internally connected routing devices, you might want your IBGP-only routers to default to the local externally connected gateway. The local IBGP-only (internal) routers skip the AS path comparison and move down the decision tree to use the closest interior gateway protocol (IGP) gateway (lowest IGP metric). Doing this might be an effective way to force these routers to use a LAN connection instead of their WAN connection.

CAUTION: **When you include the as-path-ignore statement on a routing device in your network, you might need to include it on all other BGP-enabled devices in your network to prevent routing loops and convergence issues. This is especially true for IBGP path comparisons.**

In this example, Device R2 is learning about the loopback interface address on Device R4 (4.4.4.4/32) from Device R1 and Device R3. Device R1 is advertising 4.4.4.4/32 with an AS-path of 1 5 4, and Device R3 is advertising 4.4.4.4/32 with an AS-path of 3 4. Device R2 selects the path for 4.4.4.4/32 from Device R3 as the best path because the AS path is shorter than the AS path from Device R1.

This example modifies the BGP configuration on Device R2 so that the AS-path length is not used in the best-path selection.

Device R1 has a lower router ID (1.1.1.1) than Device R3 (1.1.1.1). If all other path selection criteria are equal (or, as in this case, ignored), the route learned from Device R1 is used. Because the AS-path attribute is being ignored, the best path is toward Device R1 because of its lower router ID value.

Figure 42 on page 295 shows the sample topology.

Copyright © 2012, Juniper Networks, Inc.

Figure 42: Topology for Ignoring the AS-Path Lengh

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 192.168.10.1/24
set interfaces fe-1/2/1 unit 10 family inet address 192.168.50.2/24
set interfaces lo0 unit 1 family inet address 1.1.1.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.10.2 peer-as 2
set protocols bgp group ext neighbor 192.168.50.1 peer-as 5
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
```

```
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.20.0/24 next-hop 192.168.10.2
set routing-options static route 192.168.30.0/24 next-hop 192.168.10.2
set routing-options static route 192.168.40.0/24 next-hop 192.168.50.1
set routing-options router-id 1.1.1.1
set routing-options autonomous-system 1
```

Device R2

```
set interfaces fe-1/2/0 unit 2 family inet address 192.168.10.2/24
set interfaces fe-1/2/1 unit 3 family inet address 192.168.20.2/24
set interfaces lo0 unit 2 family inet address 2.2.2.2/32
set protocols bgp path-selection as-path-ignore
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.10.1 peer-as 1
set protocols bgp group ext neighbor 192.168.20.1 peer-as 3
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.50.0/24 next-hop 192.168.10.1
set routing-options static route 192.168.40.0/24 next-hop 192.168.10.1
set routing-options static route 192.168.30.0/24 next-hop 192.168.20.1
set routing-options router-id 2.2.2.2
set routing-options autonomous-system 2
```

Device R3

```
set interfaces fe-1/2/0 unit 4 family inet address 192.168.20.1/24
set interfaces fe-1/2/1 unit 5 family inet address 192.168.30.1/24
set interfaces lo0 unit 3 family inet address 1.1.1.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.20.2 peer-as 2
set protocols bgp group ext neighbor 192.168.30.2 peer-as 4
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.20.2
set routing-options static route 192.168.50.0/24 next-hop 192.168.20.2
set routing-options static route 192.168.40.0/24 next-hop 192.168.30.2
set routing-options router-id 3.3.3.3
set routing-options autonomous-system 3
```

Device R4

```
set interfaces fe-1/2/0 unit 6 family inet address 192.168.30.2/24
set interfaces fe-1/2/1 unit 7 family inet address 192.168.40.1/24
set interfaces lo0 unit 4 family inet address 4.4.4.4/32
```


Copyright © 2012, Juniper Networks, Inc.

```
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.30.1 peer-as 3
set protocols bgp group ext neighbor 192.168.40.2 peer-as 5
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.40.2
set routing-options static route 192.168.50.0/24 next-hop 192.168.40.2
set routing-options static route 192.168.40.0/24 next-hop 192.168.30.1
set routing-options router-id 4.4.4.4
set routing-options autonomous-system 4
```

Device R5

```
set interfaces fe-1/2/0 unit 8 family inet address 192.168.40.2/24
set interfaces fe-1/2/1 unit 9 family inet address 192.168.50.1/24
set interfaces lo0 unit 5 family inet address 5.5.5.5/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.40.1 peer-as 4
set protocols bgp group ext neighbor 192.168.50.2 peer-as 1
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.50.2
set routing-options static route 192.168.20.0/24 next-hop 192.168.50.2
set routing-options static route 192.168.30.0/24 next-hop 192.168.40.1
set routing-options router-id 5.5.5.5
set routing-options autonomous-system 5
```

Configuring Device R2

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device R2:

1. Configure the interfaces.

```
[edit interfaces]
user@R2# set fe-1/2/0 unit 2 family inet address 192.168.10.2/24
user@R2# set fe-1/2/1 unit 3 family inet address 192.168.20.2/24
user@R2# set lo0 unit 2 family inet address 2.2.2.2/32
```

2. Configure EBGP.

```
[edit protocols bgp group ext]
```

```
user@R2# set type external
user@R2# set export send-direct
user@R2# set export send-static
user@R2# set export send-local
user@R2# set neighbor 192.168.10.1 peer-as 1
user@R2# set neighbor 192.168.20.1 peer-as 3
```

3. Configure the autonomous system (AS) path attribute to be ignored in the Junos OS path selection algorithm.

```
[edit protocols bgp]
user@R2# set path-selection as-path-ignore
```

4. Configure the routing policy.

```
[edit policy-options]
user@R2# set policy-statement send-direct term 1 from protocol direct
user@R2# set policy-statement send-direct term 1 then accept
user@R2# set policy-statement send-local term 1 from protocol local
user@R2# set policy-statement send-local term 1 then accept
user@R2# set policy-statement send-static term 1 from protocol static
user@R2# set policy-statement send-static term 1 then accept
```

5. Configure some static routes.

```
[edit routing-options static]
user@R2# set route 192.168.50.0/24 next-hop 192.168.10.1
user@R2# set route 192.168.40.0/24 next-hop 192.168.10.1
user@R2# set route 192.168.30.0/24 next-hop 192.168.20.1
```

6. Configure the autonomous system (AS) number and the router ID.

```
[edit routing-options]
user@R2# set router-id 2.2.2.2
user@R2# set autonomous-system 2
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R2# show interfaces
fe-1/2/0 {
   unit 2 {
      family inet {
         address 192.168.10.2/24;
      }
   }
}
fe-1/2/1 {
   unit 3 {
      family inet {
         address 192.168.20.2/24;
      }
   }
}
lo0 {
   unit 2 {
```

 Copyright © 2012, Juniper Networks, Inc.

```
        family inet {
            address 2.2.2.2/32;
        }
    }
}
```

```
user@R2# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}
policy-statement send-local {
    term 1 {
        from protocol local;
        then accept;
    }
}
policy-statement send-static {
    term 1 {
        from protocol static;
        then accept;
    }
}
```

```
user@R2# show protocols
bgp {
    path-selection as-path-ignore;
    group ext {
        type external;
        export [ send-direct send-static send-local ];
        neighbor 192.168.10.1 {
            peer-as 1;
        }
        neighbor 192.168.20.1 {
            peer-as 3;
        }
    }
}
```

```
user@R21# show routing-options
static {
    route 192.168.50.0/24 next-hop 192.168.10.1;
    route 192.168.40.0/24 next-hop 192.168.10.1;
    route 192.168.30.0/24 next-hop 192.168.20.1;
}
router-id 2.2.2.2;
autonomous-system 2;
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat the configuration on the other devices in the network, changing the interface names and IP addresses, as needed.

Verification

Confirm that the configuration is working properly.

- Checking the Neighbor Status on page 300

Checking the Neighbor Status

Purpose Make sure that from Device R2, the active path to get to AS 4 is through AS 1 and AS 5, not through AS 3.

NOTE: To verify the functionality of the **as-path-ignore** statement, you might need to run the **restart routing** command to force reevaluation of the active path. This is because for BGP, if both paths are external, the Junos OS behavior is to prefer the currently active path. This behavior helps to minimize route-flapping. Use caution when restarting the routing protocol process in a production network.

Action From operational mode, enter the **restart routing** command.

```
user@R2> restart routing
Routing protocols process started, pid 49396
```

From operational mode, enter the **show route 4.4.4.4 protocol bgp** command.

```
user@R2> show route 4.4.4.4 protocol bgp
inet.0: 12 destinations, 25 routes (12 active, 0 holddown, 4 hidden)
+ = Active Route, - = Last Active, * = Both

4.4.4.4/32         *[BGP/170] 00:00:12, localpref 100
                      AS path: 1 5 4 I
                    > to 192.168.10.1 via fe-1/2/0.2
                    [BGP/170] 00:00:08, localpref 100
                      AS path: 3 4 I
                    > to 192.168.20.1 via fe-1/2/1.3
```

Meaning The asterisk (*) is next to the path learned from R1, meaning that this is the active path. The AS path for the active path is 1 5 4, which is longer than the AS path (3 4) for the nonactive path learned from Router R3.

Related Documentation

- Understanding BGP Path Selection on page 265

BGP Multiple Exit Discriminator

- Understanding the MED Attribute on page 301
- Example: Always Comparing MEDs on page 303

 Copyright © 2012, Juniper Networks, Inc.

Understanding the MED Attribute

The BGP multiple exit discriminator (MED, or **MULTI_EXIT_DISC**) is a non-transitive attribute, meaning that it is not propagated throughout the Internet, but only to adjacent autonomous systems (ASs). The MED attribute is optional, meaning that it is not always sent with the BGP updates. The purpose of MED is to influence how other ASs enter your AS to reach a certain prefix.

The MED attribute has a value that is referred to as a *metric*. If all other factors in determining an exit point are equal, the exit point with the lowest metric is preferred.

If a MED is received over an external BGP link, it is propagated over internal links to other BGP-enabled devices within the AS.

BGP update messages include a MED metric if the route was learned from BGP and already had a MED metric associated with it, or if you configure the MED metric in the configuration file.

A MED metric is advertised with a route according to the following general rules:

- A more specific metric overrides a less specific metric. That is, a group-specific metric overrides a global BGP metric, and a peer-specific metric overrides a global BGP or group-specific metric.
- A metric defined with a routing policy overrides a metric defined with the **metric-out** statement.
- If any metric is defined, it overrides a metric received in a route.
- If the received route does not have an associated MED metric, and if you do not explicitly configure a metric value, no metric is advertised. When you do not explicitly configure a metric value, the MED value is equivalent to zero (0) when advertising an active route.

Because the AS path rather than the number of hops between hosts is the primary criterion for BGP route selection, an AS with multiple connections to a peer AS can have multiple equivalent AS paths. When the routing table contains two routes to the same host in a neighboring AS, an MED metric assigned to each route can determine which to include in the forwarding table. The MED metric you assign can force traffic through a particular exit point in an AS.

Figure 43 on page 302 illustrates how MED metrics are used to determine route selection.

Copyright © 2012, Juniper Networks, Inc.

Figure 43: Default MED Example

Figure 43 on page 302 shows AS 1 and AS 2 connected by two separate BGP links to Routers C and D. Host E in AS 1 is located nearer to Router C. Host F, also in AS 1, is located nearer to Router D. Because the AS paths are equivalent, two routes exist for each host, one through Router C and one through Router D. To force all traffic destined for Host E through Router C, the network administrator for AS 2 assigns an MED metric for each router to Host E at its exit point. An MED metric of 10 is assigned to the route to Host E through Router C, and an MED metric of 20 is assigned to the route to Host E through Router D. BGP routers in AS 2 then select the route with the lower MED metric for the forwarding table.

By default, only the MEDs of routes that have the same peer ASs are compared. However, you can configure the routing table path selection options listed in Table 7 on page 302 to compare MEDs in different ways. The MED options are not mutually exclusive and can be configured in combination or independently. For the MED options to take effect, you must configure them uniformly all through your network. The MED option or options you configure determine the route selected. Thus we recommend that you carefully evaluate your network for preferred routes before configuring the MED options.

Table 7: MED Options for Routing Table Path Selection

Option (Name)	Function	Use
Always comparing MEDs (**always-compare-med**)	Ensures that the MEDs for paths from peers in different ASs are always compared in the route selection process.	Useful when all enterprises participating in a network agree on a uniform policy for setting MEDs. For example, in a network shared by two ISPs, both must agree that a certain path is the better path to configure the MED values correctly.

Copyright © 2012, Juniper Networks, Inc.

Table 7: MED Options for Routing Table Path Selection *(continued)*

Option (Name)	Function	Use
Adding IGP cost to MED (**med-plus-igp**)	Before comparing MED values for path selection, adds to the MED the cost of the IGP route to the BGP next-hop destination. This option replaces the MED value for the router, but does not affect the IGP metric comparison. As a result, when multiple routes have the same value after the MED-plus-IPG comparison, and route selection continues, the IGP route metric is also compared, even though it was added to the MED value and compared earlier in the selection process.	Useful when the downstream AS requires the complete cost of a certain route that is received across multiple ASs.
Applying Cisco IOS nondeterministic behavior (**cisco-non-deterministic**)	Specifies the nondeterministic behavior of the Cisco IOS software: • The active path is always first. All nonactive but eligible paths follow the active path and are maintained in the order in which they were received. Ineligible paths remain at the end of the list. • When a new path is added to the routing table, path comparisons are made among all routes, including those paths that must never be selected because they lose the MED tie-breaking rule.	We recommend that you do not configure this option, because the nondeterministic behavior sometimes prevents the system from properly comparing the MEDs between paths.

Related Documentation

- Example: Always Comparing MEDs on page 303

Example: Always Comparing MEDs

In this example, paths learned from **208.197.169.15** have their MED values compared to the sum of 4 and the MED values of the same paths learned from **208.197.169.14**:

```
[edit]
protocols {
   bgp {
      path-selection always-compare-med;
      group ref {
         type external;
         import math;
         peer-as 10458;
         neighbor 208.197.169.14;
      }
      group ref {
         type external;
         peer-as 10;
         neighbor 208.197.169.15;
```

```
        }
      }
    }
    policy-options {
      policy-statement math {
        then {
          metric add 4;
        }
      }
    }
```

Related Documentation

- Understanding the MED Attribute on page 301

BGP Route Reflectors

- Understanding BGP Route Reflectors on page 304
- Example: Configuring a Route Reflector on page 306

Understanding BGP Route Reflectors

Because of the internal BGP (IBGP) full-mesh requirement, most networks use route reflectors to simplify configuration. The formula to compute the number of sessions required for a full mesh is v * (v - 1)/2, where v is the number of BGP-enabled devices. The full-mesh model does not scale well. Using a route reflector, you group routers into clusters, which are identified by numeric identifiers unique to the autonomous system (AS). Within the cluster, you must configure a BGP session from a single router (the route reflector) to each internal peer. With this configuration, the IBGP full-mesh requirement is met.

To use route reflection in an AS, you designate one or more routers as a route reflector—typically, one per point of presence (POP). Route reflectors have the special BGP ability to readvertise routes learned from an internal peer to other internal peers. So rather than requiring all internal peers to be fully meshed with each other, route reflection requires only that the route reflector be fully meshed with all internal peers. The route reflector and all of its internal peers form a cluster, as shown in Figure 44 on page 305.

NOTE: For some Juniper Networks devices, you must have an Advanced BGP Feature license installed on each device that uses a route reflector. For license details, see the *Junos OS Initial Configuration Guide for Security Devices*.

Copyright © 2012, Juniper Networks, Inc.

Figure 44: Simple Route Reflector Topology (One Cluster)

Figure 44 on page 305 shows Router RR configured as the route reflector for Cluster 127. The other routers are designated internal peers within the cluster. BGP routes are advertised to Router RR by any of the internal peers. RR then readvertises those routes to all other peers within the cluster.

You can configure multiple clusters and link them by configuring a full mesh of route reflectors (see Figure 45 on page 305).

Figure 45: Basic Route Reflection (Multiple Clusters)

Figure 45 on page 305 shows Route Reflectors RR 1, RR 2, RR 3, and RR 4 as fully meshed internal peers. When a router advertises a route to RR 1, RR 1 readvertises the route to the other route reflectors, which, in turn, readvertise the route to the remaining routers within the AS. Route reflection allows the route to be propagated throughout the AS without the scaling problems created by the full mesh requirement.

Copyright © 2012, Juniper Networks, Inc.

However, as clusters become large, a full mesh with a route reflector becomes difficult to scale, as does a full mesh between route reflectors. To help offset this problem, you can group clusters of routers together into clusters of clusters for hierarchical route reflection (see Figure 46 on page 306).

Figure 46: Hierarchical Route Reflection (Clusters of Clusters)

Figure 46 on page 306 shows RR 2, RR 3, and RR 4 as the route reflectors for Clusters 127, 19, and 45, respectively. Rather than fully mesh those route reflectors, the network administrator has configured them as part of another cluster (Cluster 6) for which RR 1 is the route reflector. When a router advertises a route to RR 2, RR 2 readvertises the route to all the routers within its own cluster, and then readvertises the route to RR 1. RR 1 readvertises the route to the routers in its cluster, and those routers propagate the route down through their clusters.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding BGP on page 228
- Example: Configuring a Route Reflector on page 306

Example: Configuring a Route Reflector

This example shows how to configure a route reflector.

- Requirements on page 306
- Overview on page 307
- Configuration on page 308
- Verification on page 316

Requirements

No special configuration beyond device initialization is required before you configure this example.

Copyright © 2012, Juniper Networks, Inc.

Overview

Generally, internal BGP (IBGP)-enabled devices need to be fully meshed, because IBGP does not readvertise updates to other IBGP-enabled devices. The full mesh is a logical mesh achieved through configuration of multiple **neighbor** statements on each IBGP-enabled device. The full mesh is not necessarily a physical full mesh. Maintaining a full mesh (logical or physical) does not scale well in large deployments.

Figure 47 on page 308 shows an IBGP network with Device A acting as a route reflector. Device B and Device C are clients of the route reflector. Device D and Device E are outside the cluster, so they are nonclients of the route reflector.

On Device A (the route reflector), you must form peer relationships with all of the IBGP-enabled devices by including the **neighbor** statement for the clients (Device B and Device C) and the nonclients (Device D and Device E). You must also include the **cluster** statement and a cluster identifier. The cluster identifier can be any 32-bit value. This example uses the loopback interface IP address of the route reflector.

On Device B and Device C, the route reflector clients, you only need one **neighbor** statement that forms a peer relationship with the route reflector, Device A.

On Device D and Device E, the nonclients, you need a **neighbor** statement for each nonclient device (D-to-E and E-to-D). You also need a **neighbor** statement for the route reflector (D-to-A and E-to-A). Device D and Device E do not need **neighbor** statements for the client devices (Device B and Device C).

TIP: Device D and Device E are considered to be nonclients because they have explicitly configured peer relationships with each other. To make them RRroute reflector clients, remove the **neighbor 192.168.5.5** statement from the configuration on Device D, and remove the **neighbor 192.168.0.1** statement from the configuration on Device E.

Copyright © 2012, Juniper Networks, Inc.

Figure 47: IBGP Network Using a Route Reflector

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device A

```
set interfaces fe-0/0/0 unit 1 description to-B
set interfaces fe-0/0/0 unit 1 family inet address 10.10.10.1/30
set interfaces fe-0/0/1 unit 3 description to-D
set interfaces fe-0/0/1 unit 3 family inet address 10.10.10.9/30
set interfaces lo0 unit 1 family inet address 192.168.6.5/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.6.5
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers cluster 192.168.6.5
set protocols bgp group internal-peers neighbor 192.163.6.4
set protocols bgp group internal-peers neighbor 192.168.40.4
set protocols bgp group internal-peers neighbor 192.168.0.1
set protocols bgp group internal-peers neighbor 192.168.5.5
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.1
set protocols ospf area 0.0.0.0 interface fe-0/0/1.3
set policy-options policy-statement send-ospf term 2 from protocol ospf
```


Copyright © 2012, Juniper Networks, Inc.

```
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.6.5
set routing-options autonomous-system 17
```

Device B

```
set interfaces fe-0/0/0 unit 2 description to-A
set interfaces fe-0/0/0 unit 2 family inet address 10.10.10.2/30
set interfaces fe-0/0/1 unit 5 description to-C
set interfaces fe-0/0/1 unit 5 family inet address 10.10.10.5/30
set interfaces lo0 unit 2 family inet address 192.163.6.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.163.6.4
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.2
set protocols ospf area 0.0.0.0 interface fe-0/0/1.5
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.163.6.4
set routing-options autonomous-system 17
```

Device C

```
set interfaces fe-0/0/0 unit 6 description to-B
set interfaces fe-0/0/0 unit 6 family inet address 10.10.10.6/30
set interfaces lo0 unit 3 family inet address 192.168.40.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.40.4
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.6
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.40.4
set routing-options autonomous-system 17
```

Device D

```
set interfaces fe-0/0/0 unit 4 description to-A
set interfaces fe-0/0/0 unit 4 family inet address 10.10.10.10/30
set interfaces fe-0/0/1 unit 7 description to-E
set interfaces fe-0/0/1 unit 7 family inet address 10.10.10.13/30
set interfaces lo0 unit 4 family inet address 192.168.0.1/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.0.1
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols bgp group internal-peers neighbor 192.168.5.5
set protocols ospf area 0.0.0.0 interface lo0.4 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.4
set protocols ospf area 0.0.0.0 interface fe-0/0/1.7
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 17
```

Device E

```
set interfaces fe-0/0/0 unit 8 description to-D
set interfaces fe-0/0/0 unit 8 family inet address 10.10.10.14/30
```

Copyright © 2012, Juniper Networks, Inc.

```
set interfaces lo0 unit 5 family inet address 192.168.5.5/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.5.5
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.0.1
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.5 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.8
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.5.5
set routing-options autonomous-system 17
```

Configuring the Route Reflector

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure IBGP in the network using Juniper Networks Device A as a route reflector:

1. Configure the interfaces.

```
[edit interfaces]
user@A# set fe-0/0/0 unit 1 description to-B
user@A# set fe-0/0/0 unit 1 family inet address 10.10.10.1/30
user@A# set fe-0/0/1 unit 3 description to-D
user@A# set fe-0/0/1 unit 3 family inet address 10.10.10.9/30
user@A# set lo0 unit 1 family inet address 192.168.6.5/32
```

2. Configure BGP, including the cluster identifier and neighbor relationships with all IBGP-enabled devices in the autonomous system (AS).

   Also apply the policy that redistributes OSPF routes into BGP.

```
[edit protocols bgp group internal-peers]
user@A# set type internal
user@A# set local-address 192.168.6.5
user@A# set export send-ospf
user@A# set cluster 192.168.6.5
user@A# set neighbor192.163.6.4
user@A# set neighbor 192.168.40.4
user@A# set neighbor 192.168.0.1
user@A# set neighbor 192.168.5.5
```

3. Configure static routing or an interior gateway protocol (IGP).

   This example uses OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@A# set interface lo0.1 passive
user@A# set interface fe-0/0/0.1
user@A# set interface fe-0/0/1.3
```

4. Configure the policy that redistributes OSPF routes into BGP.

```
[edit policy-options policy-statement send-ospf term 2]
user@A# set from protocol ospf
user@A# set then accept
```

 Copyright © 2012, Juniper Networks, Inc.

5. Configure the router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@A# set router-id 192.168.6.5
user@A# set autonomous-system 17
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@A# show interfaces
fe-0/0/0 {
   unit 1 {
      description to-B;
      family inet {
         address 10.10.10.1/30;
      }
   }
}
fe-0/0/1 {
   unit 3 {
      description to-D;
      family inet {
         address 10.10.10.9/30;
      }
   }
}
lo0 {
   unit 1 {
      family inet {
         address 192.168.6.5/32;
      }
   }
}
```

```
user@A# show protocols
bgp {
   group internal-peers {
      type internal;
      local-address 192.168.6.5;
      export send-ospf;
      cluster 192.168.6.5;
      neighbor 192.163.6.4;
      neighbor 192.168.40.4;
      neighbor 192.168.0.1;
      neighbor 192.168.5.5;
   }
}
ospf {
   area 0.0.0.0 {
      interface lo0.1 {
         passive;
      }
      interface fe-0/0/0.1;
      interface fe-0/0/1.3;
```

```
    }
  }

  user@A# show policy-options
  policy-statement send-ospf {
    term 2 {
      from protocol ospf;
      then accept;
    }
  }

  user@A# show routing-options
  router-id 192.168.6.5;
  autonomous-system 17;
```

If you are done configuring the device, enter **commit** from configuration mode.

NOTE: Repeat these steps for each nonclient BGP peer within the cluster that you are configuring, if the other nonclient devices are from Juniper Networks. Otherwise, consult the device's documentation for instructions.

Configuring Client Peers

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure client peers:

1. Configure the interfaces.

```
[edit interfaces]
user@B# set fe-0/0/0 unit 2 description to-A
user@B# set fe-0/0/0 unit 2 family inet address 10.10.10.2/30
user@B# set fe-0/0/1 unit 5 description to-C
user@B# set fe-0/0/1 unit 5 family inet address 10.10.10.5/30
user@B# set lo0 unit 2 family inet address 192.163.6.4/32
```

2. Configure the BGP neighbor relationship with the route reflector.

   Also apply the policy that redistributes OSPF routes into BGP.

```
[edit protocols bgp group internal-peers]
user@B# set type internal
user@B# set local-address 192.163.6.4
user@B# set export send-ospf
user@B# set neighbor 192.168.6.5
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@B# set interface lo0.2 passive
user@B# set interface fe-0/0/0.2
user@B# set interface fe-0/0/1.5
```

4. Configure the policy that redistributes OSPF routes into BGP.

Copyright © 2012, Juniper Networks, Inc.

```
[edit policy-options policy-statement send-ospf term 2]
user@B# set from protocol ospf
user@B# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@B# set router-id 192.163.6.4
user@B# set autonomous-system 17
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@B# show interfaces
fe-0/0/0 {
   unit 2 {
      description to-A;
      family inet {
         address 10.10.10.2/30;
      }
   }
}
fe-0/0/1 {
   unit 5 {
      description to-C;
      family inet {
         address 10.10.10.5/30;
      }
   }
}
lo0 {
   unit 2 {
      family inet {
         address 192.163.6.4/32;
      }
   }
}
```

```
user@B# show protocols
bgp {
   group internal-peers {
      type internal;
      local-address 192.163.6.4;
      export send-ospf;
      neighbor 192.168.6.5;
   }
}
ospf {
   area 0.0.0.0 {
      interface lo0.2 {
         passive;
      }
      interface fe-0/0/0.2;
      interface fe-0/0/1.5;
```

```
    }
  }

  user@B# show policy-options
  policy-statement send-ospf {
    term 2 {
      from protocol ospf;
      then accept;
    }
  }

  user@B# show routing-options
  router-id 192.163.6.4;
  autonomous-system 17;
```

If you are done configuring the device, enter **commit** from configuration mode.

NOTE: Repeat these steps for each client BGP peer within the cluster that you are configuring if the other client devices are from Juniper Networks. Otherwise, consult the device's documentation for instructions.

Configuring Nonclient Peers

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure nonclient peers:

1. Configure the interfaces.

```
[edit interfaces]
user@D# set fe-0/0/0 unit 4 description to-A
user@D# set fe-0/0/0 unit 4 family inet address 10.10.10.10/30
user@D# set fe-0/0/1 unit 7 description to-E
user@D# set fe-0/0/1 unit 7 family inet address 10.10.10.13/30
user@D# set lo0 unit 4 family inet address 192.168.0.1/32
```

2. Configure the BGP neighbor relationships with the RRroute reflector and with the other nonclient peers.

   Also apply the policy that redistributes OSPF routes into BGP.

```
[edit protocols bgp group internal-peers]
user@D# set type internal
user@D# set local-address 192.168.0.1
user@D# set export send-ospf
user@D# set neighbor 192.168.6.5
user@D# set neighbor 192.168.5.5
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@D# set interface lo0.4 passive
user@D# set interface fe-0/0/0.4
```


Copyright © 2012, Juniper Networks, Inc.

```
user@D# set interface fe-0/0/1.7
```

4. Configure the policy that redistributes OSPF routes into BGP.

```
[edit policy-options policy-statement send-ospf term 2]
user@D# set from protocol ospf
user@D# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@D# set router-id 192.168.0.1
user@D# set autonomous-system 17
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@D# show interfaces
fe-0/0/0 {
   unit 4 {
      description to-A;
      family inet {
         address 10.10.10.10/30;
      }
   }
}
fe-0/0/1 {
   unit 7 {
      description to-E;
      family inet {
         address 10.10.10.13/30;
      }
   }
}
lo0 {
   unit 4 {
      family inet {
         address 192.168.0.1/32;
      }
   }
}

user@D# show protocols
bgp {
   group internal-peers {
      type internal;
      local-address 192.168.0.1;
      export send-ospf;
      neighbor 192.168.6.5;
      neighbor 192.168.5.5;
   }
}
ospf {
   area 0.0.0.0 {
      interface lo0.4 {
```

```
            passive;
        }
        interface fe-0/0/0.4;
        interface fe-0/0/1.7;
    }
}
```

```
user@D# show policy-options
policy-statement send-ospf {
    term 2 {
        from protocol ospf;
        then accept;
    }
}
```

```
user@D# show routing-options
router-id 192.168.0.1;
autonomous-system 17;
```

If you are done configuring the device, enter **commit** from configuration mode.

NOTE: Repeat these steps for each nonclient BGP peer within the cluster that you are configuring if the other nonclient devices are from Juniper Networks. Otherwise, consult the device's documentation for instructions.

Verification

Confirm that the configuration is working properly.

- Verifying BGP Neighbors on page 316
- Verifying BGP Groups on page 319
- Verifying BGP Summary Information on page 319
- Verifying Routing Table Information on page 319

Verifying BGP Neighbors

Purpose Verify that BGP is running on configured interfaces and that the BGP session is established for each neighbor address.

Action From operational mode, enter the **show bgp neighbor** command.

```
user@A> show bgp neighbor
Peer: 192.163.6.4+179 AS 17    Local: 192.168.6.5+62857 AS 17
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.163.6.4     Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
```

 Copyright © 2012, Juniper Networks, Inc.

```
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            6
    Accepted prefixes:            1
    Suppressed due to damping:    0
    Advertised prefixes:          6
  Last traffic (seconds): Received 5    Sent 3    Checked 19
  Input messages:  Total 2961   Updates 7       Refreshes 0     Octets 56480
  Output messages: Total 2945   Updates 6       Refreshes 0     Octets 56235
  Output Queue[0]: 0

Peer: 192.168.0.1+179 AS 17    Local: 192.168.6.5+60068 AS 17
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.0.1     Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 3
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            6
    Accepted prefixes:            1
    Suppressed due to damping:    0
    Advertised prefixes:          6
  Last traffic (seconds): Received 18   Sent 20   Checked 12
  Input messages:  Total 15     Updates 5       Refreshes 0     Octets 447
  Output messages: Total 554    Updates 4       Refreshes 0     Octets 32307
```

```
  Output Queue[0]: 0

Peer: 192.168.5.5+57458 AS 17  Local: 192.168.6.5+179 AS 17
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.5.5     Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 2
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            7
    Accepted prefixes:            7
    Suppressed due to damping:    0
    Advertised prefixes:          6
  Last traffic (seconds): Received 17   Sent 3    Checked 9
  Input messages:  Total 2967   Updates 7       Refreshes 0     Octets 56629
  Output messages: Total 2943   Updates 6       Refreshes 0     Octets 56197
  Output Queue[0]: 0

Peer: 192.168.40.4+53990 AS 17 Local: 192.168.6.5+179 AS 17
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.40.4    Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
```

 Copyright © 2012, Juniper Networks, Inc.

```
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            7
    Accepted prefixes:            7
    Suppressed due to damping:    0
    Advertised prefixes:          6
  Last traffic (seconds): Received 5    Sent 23   Checked 52
  Input messages:  Total 2960   Updates 7       Refreshes 0     Octets 56496
  Output messages: Total 2943   Updates 6       Refreshes 0     Octets 56197
  Output Queue[0]: 0
```

Verifying BGP Groups

Purpose Verify that the BGP groups are configured correctly.

Action From operational mode, enter the **show bgp group** command.

```
user@A> show bgp group
Group Type: Internal    AS: 17                      Local AS: 17
  Name: internal-peers  Index: 0                    Flags: <>
  Export: [ send-ospf ]
  Options: <Cluster>
  Holdtime: 0
  Total peers: 4        Established: 4
  192.163.6.4+179
  192.168.40.4+53990
  192.168.0.1+179
  192.168.5.5+57458
  inet.0: 0/26/16/0

Groups: 1  Peers: 4    External: 0    Internal: 4    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                26          0          0          0          0          0
```

Verifying BGP Summary Information

Purpose Verify that the BGP configuration is correct.

Action From operational mode, enter the **show bgp summary** command.

```
user@A> show bgp summary

Groups: 1 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                26          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.163.6.4              17       2981       2965       0       0    22:19:15 0/6/1/0              0/0/0/0
192.168.0.1              17         36        575       0       0       13:43 0/6/1/0              0/0/0/0
192.168.5.5              17       2988       2964       0       0    22:19:10 0/7/7/0              0/0/0/0
192.168.40.4             17       2980       2964       0       0    22:19:14 0/7/7/0              0/0/0/0
```

Verifying Routing Table Information

Purpose Verify that the routing table contains the IBGP routes.

Action From operational mode, enter the **show route** command.

```
user@A> show route
inet.0: 12 destinations, 38 routes (12 active, 0 holddown, 10 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30      *[Direct/0] 22:22:03
                    > via fe-0/0/0.1
                    [BGP/170] 22:20:55, MED 2, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:51, MED 3, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
10.10.10.1/32      *[Local/0] 22:22:03
                      Local via fe-0/0/0.1
10.10.10.4/30      *[OSPF/10] 22:21:13, metric 2
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:51, MED 4, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
10.10.10.8/30      *[Direct/0] 22:22:03
                    > via fe-0/0/1.3
                    [BGP/170] 22:20:51, MED 2, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:55, MED 3, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
10.10.10.9/32      *[Local/0] 22:22:03
                      Local via fe-0/0/1.3
10.10.10.12/30     *[OSPF/10] 22:21:08, metric 2
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:55, MED 4, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
192.163.6.4/32     *[OSPF/10] 22:21:13, metric 1
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:55, MED 1, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:51, MED 3, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
192.168.0.1/32     *[OSPF/10] 22:21:08, metric 1
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:51, MED 1, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:55, MED 3, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
192.168.5.5/32     *[OSPF/10] 22:21:08, metric 2
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 00:15:24, MED 1, localpref 100, from 192.168.0.1
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:55, MED 4, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
192.168.6.5/32     *[Direct/0] 22:22:04
```

 Copyright © 2012, Juniper Networks, Inc.

```
                    > via lo0.1
                    [BGP/170] 22:20:51, MED 2, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:55, MED 2, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
192.168.40.4/32    *[OSPF/10] 22:21:13, metric 2
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:55, MED 1, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:51, MED 4, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
224.0.0.5/32       *[OSPF/10] 22:22:07, metric 1
                      MultiRecv
```

Related Documentation

- *Junos OS Policy Framework Configuration Guide*
- Example: Configuring Internal BGP Peer Sessions on page 254
- Example: Configuring External BGP Point-to-Point Peer Sessions on page 247
- Understanding BGP Route Reflectors on page 304
- BGP Configuration Overview on page 232

BGP Confederations

- Understanding BGP Confederations on page 321
- Example: Configuring BGP Confederations on page 322

Understanding BGP Confederations

BGP confederations are another way to solve the scaling problems created by the BGP full mesh requirement. BGP confederations effectively break up a large autonomous system (AS) into subautonomous systems (sub-ASs). Each sub-AS must be uniquely identified within the confederation AS by a sub-AS number. Typically, sub-AS numbers are taken from the private AS numbers between 64,512 and 65,535.

Within a sub-AS, the same internal BGP (IBGP) full mesh requirement exists. Connections to other confederations are made with standard external BGP (EBGP), and peers outside the sub-AS are treated as external. To avoid routing loops, a sub-AS uses a confederation sequence, which operates like an AS path but uses only the privately assigned sub-AS numbers.

The confederation AS appears whole to other confederation ASs. The AS path received by other ASs shows only the globally assigned AS number. It does not include the confederation sequence or the privately assigned sub-AS numbers. The sub-AS numbers are removed when the route is advertised out of the confederation AS. Figure 48 on page 322 shows an AS divided into four confederations.

Figure 48: BGP Confederations

Figure 48 on page 322 shows AS 3 divided into four sub-ASs, 64517, 64550, 65300, and 65410, which are linked through EBGP sessions. Because the confederations are connected by EBGP, they do not need to be fully meshed. EBGP routes are readvertised to other sub-ASs.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding BGP on page 228
- Example: Configuring BGP Confederations on page 322

Example: Configuring BGP Confederations

This example shows how to configure BGP confederations.

- Requirements on page 322
- Overview on page 323
- Configuration on page 323
- Verification on page 325

Requirements

- Configure network interfaces.
- Configure external peer sessions. See "Example: Configuring External BGP Point-to-Point Peer Sessions" on page 247.
- Configure interior gateway protocol (IGP) sessions between peers.
- Configure a routing policy to advertise the BGP routes.

Copyright © 2012, Juniper Networks, Inc.

Overview

Within a BGP confederation, the links between the confederation member autonomous systems (ASs) must be external BGP (EBGP) links, not internal BGP (IBGP) links.

Similar to route reflectors, BGP confederations reduce the number of peer sessions and TCP sessions to maintain connections between IBGP routing devices. BGP confederation is one method used to solve the scaling problems created by the IBGP full mesh requirement. BGP confederations effectively break up a large AS into subautonomous systems. Each sub-AS must be uniquely identified within the confederation AS by a sub-AS number. Typically, sub-AS numbers are taken from the private AS numbers between 64512 and 65535. Within a sub-AS, the same IBGP full mesh requirement exists. Connections to other confederations are made with standard EBGP, and peers outside the sub-AS are treated as external. To avoid routing loops, a sub-AS uses a confederation sequence, which operates like an AS path but uses only the privately assigned sub-AS numbers.

Figure 49 on page 323 shows a sample network in which AS 17 has two separate confederations: sub-AS 64512 and sub-AS 64513, each of which has multiple routers. Within a sub-AS, an IGP is used to establish network connectivity with internal peers. Between sub-ASs, an EBGP peer session is established.

Figure 49: Typical Network Using BGP Confederations

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

All Devices in Sub-AS 64512

```
set routing-options autonomous-system 64512
set routing-options confederation 17 members 64512
set routing-options confederation 17 members 64513
set protocols bgp group sub-AS-64512 type internal
set protocols bgp group sub-AS-64512 local-address 192.168.5.1
set protocols bgp group sub-AS-64512 neighbor 192.168.8.1
set protocols bgp group sub-AS-64512 neighbor 192.168.15.1
```

Border Device in Sub-AS 64512

```
set protocols bgp group to-sub-AS-64513 type external
set protocols bgp group to-sub-AS-64513 peer-as 64513
set protocols bgp group to-sub-AS-64513 neighbor 192.168.5.2
```

All Devices in Sub-AS 64513

```
set routing-options autonomous-system 64513
set routing-options confederation 17 members 64512
set routing-options confederation 17 members 64513
set protocols bgp group sub-AS-64513 type internal
set protocols bgp group sub-AS-64513 local-address 192.168.5.2
set protocols bgp group sub-AS-64513 neighbor 192.168.9.1
set protocols bgp group sub-AS-64513 neighbor 192.168.16.1
```

Border Device in Sub-AS 64513

```
set protocols bgp group to-sub-AS-64512 type external
set protocols bgp group to-sub-AS-64512 peer-as 64512
set protocols bgp group to-sub-AS-64512 neighbor 192.168.5.1
```

Step-by-Step Procedure

This procedure shows the steps for the devices that are in sub-AS 64512.

The **autonomous-system** statement sets the sub-AS number of the device.

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure BGP confederations:

1. Set the sub-AS number for the device.

   ```
   [edit routing-options]
   user@host# set autonomous-system 64512
   ```

2. In the confederation, include all sub-ASs in the main AS.

   The number 17 represents the main AS. The **members** statement lists all the sub-ASs in the main AS.

   ```
   [edit routing-options confederation]
   user@host# set 17 members 64512
   user@host# set 17 members 64513
   ```

3. On the border device in sub-AS 64512, configure an EBGP connection to the border device in AS 64513.

   ```
   [edit protocols bgp group to-sub-AS-64513]
   user@host# set type external
   user@host# set neighbor 192.168.5.2
   user@host# set peer-as 64513
   ```

4. Configure an IBGP group for peering with the devices within sub-AS 64512.

   ```
   [edit protocols bgp group sub-AS-64512]
   user@host# set type internal
   user@host# set local-address 192.168.5.1
   user@host# neighbor 192.168.8.1
   user@host# neighbor 192.168.15.1
   ```


Copyright © 2012, Juniper Networks, Inc.

Results From configuration mode, confirm your configuration by entering the **show routing-options** and **show protocols** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show routing-options
autonomous-system 64512;
confederation 17 members [ 64512 64513 ];

user@host# show protocols
bgp {
  group to-sub-AS-64513 { # On the border devices only
    type external;
    peer-as 64513;
    neighbor 192.168.5.2;
  }
  group sub-AS-64512 {
    type internal;
    local-address 192.168.5.1;
    neighbor 192.168.8.1;
    neighbor 192.168.15.1;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.
Repeat these steps for sSub-AS 64513.

Verification

Confirm that the configuration is working properly.

- Verifying BGP Neighbors on page 325
- Verifying BGP Groups on page 326
- Verifying BGP Summary Information on page 327

Verifying BGP Neighbors

Purpose Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

Action From the CLI, enter the **show bgp neighbor** command.

Sample Output

```
user@host> show bgp neighbor
Peer: 10.255.245.12+179 AS 35  Local: 10.255.245.13+2884 AS 35
  Type: Internal    State: Established  (route reflector client)Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: Preference LocalAddress HoldTime Cluster AddressFamily Rib-group Refresh

  Address families configured: inet-vpn-unicast inet-labeled-unicast
  Local Address: 10.255.245.13 Holdtime: 90 Preference: 170
  Flags for NLRI inet-vpn-unicast: AggregateLabel
  Flags for NLRI inet-labeled-unicast: AggregateLabel
  Number of flaps: 0
  Peer ID: 10.255.245.12    Local ID: 10.255.245.13    Active Holdtime: 90
```

```
  Keepalive Interval: 30
  NLRI advertised by peer: inet-vpn-unicast inet-labeled-unicast
  NLRI for this session: inet-vpn-unicast inet-labeled-unicast
  Peer supports Refresh capability (2)
Restart time configured on the peer: 300
  Stale routes from peer are kept for: 60
  Restart time requested by this peer: 300
  NLRI that peer supports restart for: inet-unicast inet6-unicast
  NLRI that restart is negotiated for: inet-unicast inet6-unicast
  NLRI of received end-of-rib markers: inet-unicast inet6-unicast
  NLRI of all end-of-rib markers sent: inet-unicast inet6-unicast
  Table inet.0 Bit: 10000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 4
    Received prefixes: 6
    Suppressed due to damping: 0
  Table inet6.0 Bit: 20000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 0
    Received prefixes: 2
    Suppressed due to damping: 0
  Last traffic (seconds): Received 3    Sent 3    Checked 3
  Input messages:  Total 9      Updates 6       Refreshes 0     Octets 403
  Output messages: Total 7      Updates 3       Refreshes 0     Octets 365
  Output Queue[0]: 0
  Output Queue[1]: 0
  Trace options: detail packets
  Trace file: /var/log/bgpgr size 131072 files 10
```

Meaning The output shows a list of the BGP neighbors with detailed session information. Verify the following information:

- Each configured peering neighbor is listed.
- For **State**, each BGP session is **Established**.
- For **Type**, each peer is configured as the correct type (either internal or external).
- For **AS**, the AS number of the BGP neighbor is correct.

Verifying BGP Groups

Purpose Verify that the BGP groups are configured correctly.

Action From the CLI, enter the **show bgp group** command.

Sample Output

```
user@host> show bgp group
Group Type: Internal    AS: 10045       Local AS: 10045
  Name: pe-to-asbr2                                     Flags: Export Eval
  Export: [ match-all ]
  Total peers: 1        Established: 1
  10.0.0.4+179
  bgp.l3vpn.0: 1/1/0
  vpn-green.inet.0: 1/1/0
```

 Copyright © 2012, Juniper Networks, Inc.

```
Groups: 1   Peers: 1    External: 0    Internal: 1    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            1          1          0          0          0          0
```

Meaning The output shows a list of the BGP groups with detailed group information. Verify the following information:

- Each configured group is listed.
- For **AS**, each group's remote AS is configured correctly.
- For **Local AS**, each group's local AS is configured correctly.
- For **Group Type**, each group has the correct type (either internal or external).
- For **Total peers**, the expected number of peers within the group is shown.
- For **Established**, the expected number of peers within the group have BGP sessions in the **Established** state.
- The IP addresses of all the peers within the group are present.

Verifying BGP Summary Information

Purpose Verify that the BGP configuration is correct.

Action From the CLI, enter the **show bgp summary** command.

Sample Output

```
user@host> show bgp summary
Groups: 1 Peers: 3 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 6          4          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.0.0.2        65002      88675      88652       0       2       42:38 2/4/0
            0/0/0
10.0.0.3        65002      54528      54532       0       1     2w4d22h 0/0/0
            0/0/0
10.0.0.4        65002      51597      51584       0       0     2w3d22h 2/2/0
            0/0/0
```

Meaning The output shows a summary of BGP session information. Verify the following information:

- For **Groups**, the total number of configured groups is shown.
- For **Peers**, the total number of BGP peers is shown.
- For **Down Peers**, the total number of unestablished peers is 0. If this value is not zero, one or more peering sessions are not yet established.
- Under **Peer**, the IP address for each configured peer is shown.
- Under **AS**, the peer AS for each configured peer is correct.
- Under **Up/Dwn State**, the BGP state reflects the number of paths received from the neighbor, the number of these paths that have been accepted, and the number of

routes being damped (such as 0/0/0). If the field is **Active**, it indicates a problem in the establishment of the BGP session.

Related Documentation

- *Junos OS Policy Framework Configuration Guide*
- Understanding BGP Confederations on page 321
- BGP Configuration Overview on page 232

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 12

BFD

- BFD and Static Routes on page 329
- BFD Authentication and Static Routes on page 344

BFD and Static Routes

- Understanding BFD for Static Routes on page 329
- Example: Configuring BFD for Static Routes on page 332
- Example: Enabling BFD on Qualified Next Hops in Static Routes on page 338

Understanding BFD for Static Routes

The Bidirectional Forwarding Detection (BFD) protocol is a simple hello mechanism that detects failures in a network. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. BFD works with a wide variety of network environments and topologies. The failure detection timers for BFD have shorter time limits than the failure detection mechanisms of static routes, providing faster detection. These timers are also adaptive. For example, a timer can adapt to a higher value if an adjacency fails, or a neighbor can negotiate a higher value than the one configured. By default, BFD is supported on single-hop static routes. In Junos OS Release 8.2 and later, BFD also supports multihop static routes.

To enable failure detection, include the **bfd-liveness-detection** statement in the static route configuration.

In Junos OS Release 9.1 and later, the BFD protocol is supported for IPv6 static routes. Global unicast and link-local IPv6 addresses are supported for static routes. The BFD protocol is not supported on multicast or anycast IPv6 addresses. For IPv6, the BFD protocol supports only static routes and only in Junos OS Release 9.3 and later. IPv6 for BFD is not supported for any other protocol.

To configure the BFD protocol for IPv6 static routes, include the **bfd-liveness-detection** statement at the **[edit routing-options rib inet6.0 static route *destination-prefix*]** hierarchy level.

In Junos OS Release 8.5 and later, you can configure a hold-down interval to specify how long the BFD session must remain up before state change notification is sent.

Copyright © 2012, Juniper Networks, Inc.

To specify the hold-down interval, include the **holddown-interval** statement in the BFD configuration.

You can configure a number in the range from 0 through 255,000 milliseconds, and the default is 0. If the BFD session goes down and then comes back up during the hold-down interval, the timer is restarted.

NOTE: If a single BFD session includes multiple static routes, the hold-down interval with the highest value is used.

To specify the minimum transmit and receive intervals for failure detection, include the **minimum-interval** statement in the BFD configuration.

This value represents the minimum interval at which the local routing device transmits hello intervals as well as the minimum interval that the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a number in the range from 1 through 255,000 milliseconds. You can also specify the minimum transmit and receive intervals separately.

NOTE: BFD is an intensive protocol that consumes system resources. Specifying a minimum interval for BFD less than 100 ms for Routing Engine-based sessions and 10 ms for distributed BFD sessions can cause undesired BFD flapping.

Depending on your network environment, these additional recommendations might apply:

- For large-scale network deployments with a large number of BFD sessions, specify a minimum interval of 300 ms for Routing Engine-based sessions and 100 ms for distributed BFD sessions.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing (NSR) is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with NSR configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

To specify only the minimum receive interval for failure detection, include the **minimum-receive-interval** statement in the BFD configuration.

This value represents the minimum interval at which the local routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a number in the range from 1 through 255,000 milliseconds.

Copyright © 2012, Juniper Networks, Inc.

To specify the number of hello packets not received by the neighbor that causes the originating interface to be declared down, include the **multiplier** statement in the BFD configuration.

The default value is 3. You can configure a number in the range from 1 through 255.

To specify a threshold for detecting the adaptation of the detection time, include the **threshold** statement in the BFD configuration.

When the BFD session detection time adapts to a value equal to or higher than the threshold, a single trap and a system log message are sent. The detection time is based on the multiplier of the **minimum-interval** or the **minimum-receive-interval** value. The threshold must be a higher value than the multiplier for either of these configured values. For example if the **minimum-receive-interval** is 300 ms and the **multiplier** is 3, the total detection time is 900 ms. Therefore, the detection time threshold must have a value higher than 900.

To specify only the minimum transmit interval for failure detection, include the **transmit-interval minimum-interval** statement in the BFD configuration.

This value represents the minimum interval at which the local routing device transmits hello packets to the neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds.

To specify the transmit threshold for detecting the adaptation of the transmit interval, include the **transmit-interval threshold** statement in the BFD configuration.

The threshold value must be greater than the transmit interval. When the BFD session detection time adapts to a value higher than the threshold, a single trap and a system log message are sent. The detection time is based on the multiplier of the **minimum-interval** or the **minimum-receive-interval** value. The threshold must be a higher value than the multiplier for either of these configured values.

To specify the BFD version, include the **version** statement in the BFD configuration.

The default is to have the version detected automatically.

To include an IP address for the next hop of the BFD session, include the **neighbor** statement in the BFD configuration.

NOTE: **You must configure the neighbor statement if the next hop specified is an interface name. If you specify an IP address as the next hop, that address is used as the neighbor address for the BFD session.**

In Junos OS Release 9.0 and later, you can configure BFD sessions not to adapt to changing network conditions.

To disable BFD adaptation, include the **no-adaptation** statement in the BFD configuration.

NOTE: We recommend that you not disable BFD adaptation unless it is preferable not to have BFD adaptation in your network.

NOTE: If BFD is configured only on one end of a static route, the route is removed from the routing table. BFD establishes a session when BFD is configured on both ends of the static route.

BFD is not supported on ISO address families in static routes. BFD does support IS-IS.

If you configure graceful Routing Engine switchover (GRES) at the same time as BFD, GRES does not preserve the BFD state information during a failover.

Junos OS also supports BFD over multihop static routes. For example, you can configure BFD over a Layer 3 path to provide path integrity over that path. You can limit the number of hops by specifying the time-to-live (TTL).

To configure BFD over multihop static routes, include the following statements:

```
static route destination-prefix {
  bfd-liveness-detection {
    local-address ip-address;
    minimum-receive-ttl number;
  }
}
```

To specify the source address for the multihop static route and to enable multihop BFD support, include the **local-address** statement.

To specify the number of hops, include the **minimum-receive-ttl** statement. You must configure this statement for a multihop BFD session. You can configure a value in the range from 1 through 255. It is optional for a single-hop BFD session. If you configure the **minimum-receive-ttl** statement for a single-hop session, the value must be 255.

Related Documentation

- Example: Configuring BFD for Static Routes on page 332
- Example: Enabling BFD on Qualified Next Hops in Static Routes on page 338

Example: Configuring BFD for Static Routes

This example shows how to configure Bidirectional Forwarding Detection (BFD) for static routes.

- Requirements on page 333
- Overview on page 333
- Configuration on page 333
- Verification on page 336

 Copyright © 2012, Juniper Networks, Inc.

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

There are many practical applications for static routes. Static routing is often used at the network edge to support attachment to stub networks, which, given their single point of entry and egress, are well suited to the simplicity of a static route. In Junos OS, static routes have a global preference of 5. Static routes are activated if the specified next hop is reachable.

In this example, you configure the static route 192.168.47.0/24 from the provider network to the customer network, using the next-hop address of 172.16.1.2. You also configure a static default route of 0.0.0.0/0 from the customer network to the provider network, using a next-hop address of 172.16.1.1.

For demonstration purposes, some loopback interfaces are configured on Device B and Device D. These loopback interfaces provide addresses to ping and thus verify that the static routes are working.

Figure 50 on page 333 shows the sample network.

Figure 50: Customer Routes Connected to a Service Provider

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device B

```
set interfaces ge-1/2/0 unit 0 description B->D
set interfaces ge-1/2/0 unit 0 family inet address 172.16.1.1/24
```

```
set interfaces lo0 unit 57 family inet address 10.0.0.1/32
set interfaces lo0 unit 57 family inet address 10.0.0.2/32
set routing-options static route 192.168.47.0/24 next-hop 172.16.1.2
set routing-options static route 192.168.47.0/24 bfd-liveness-detection minimum-interval
  1000
set protocols bfd traceoptions file bfd-trace
set protocols bfd traceoptions flag all
```

Device D

```
set interfaces ge-1/2/0 unit 1 description D->B
set interfaces ge-1/2/0 unit 1 family inet address 172.16.1.2/24
set interfaces lo0 unit 2 family inet address 192.168.47.5/32
set interfaces lo0 unit 2 family inet address 192.168.47.6/32
set routing-options static route 0.0.0.0/0 next-hop 172.16.1.1
set routing-options static route 0.0.0.0/0 bfd-liveness-detection minimum-interval 1000
set protocols bfd traceoptions file bfd-trace
set protocols bfd traceoptions flag all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure BFD for static routes:

1. On Device B, configure the interfaces.

   ```
   [edit interfaces]
   user@B# set ge-1/2/0 unit 0 description B->D
   user@B# set ge-1/2/0 unit 0 family inet address 172.16.1.1/24
   user@B# set lo0 unit 57 family inet address 10.0.0.1/32
   user@B# set lo0 unit 57 family inet address 10.0.0.2/32
   ```

2. On Device B, create a static route and set the next-hop address.

   ```
   [edit routing-options]
   user@B# set static route 192.168.47.0/24 next-hop 172.16.1.2
   ```

3. On Device B, configure BFD for the static route.

   ```
   [edit routing-options]
   user@B# set static route 192.168.47.0/24 bfd-liveness-detection minimum-interval
     1000
   ```

4. On Device B, configure tracing operations for BFD.

   ```
   [edit protocols]
   user@B# set bfd traceoptions file bfd-trace
   user@B# set bfd traceoptions flag all
   ```

5. If you are done configuring Device B, commit the configuration.

   ```
   [edit]
   user@B# commit
   ```

6. On Device D, configure the interfaces.

   ```
   [edit interfaces]
   user@D# set ge-1/2/0 unit 1 description D->B
   user@D# set ge-1/2/0 unit 1 family inet address 172.16.1.2/24
   user@D# set lo0 unit 2 family inet address 192.168.47.5/32
   ```


Copyright © 2012, Juniper Networks, Inc.

```
user@D# set lo0 unit 2 family inet address 192.168.47.6/32
```

7. On Device D, create a static route and set the next-hop address.

```
[edit routing-options]
user@D# set static route 0.0.0.0/0 next-hop 172.16.1.1
```

8. On Device D, configure BFD for the static route.

```
[edit routing-options]
user@D# set static route 0.0.0.0/0 bfd-liveness-detection minimum-interval 1000
```

9. On Device D, configure tracing operations for BFD.

```
[edit protocols]
user@D# set bfd traceoptions file bfd-trace
user@D# set bfd traceoptions flag all
```

10. If you are done configuring Device D, commit the configuration.

```
[edit]
user@D# commit
```

Results Confirm your configuration by issuing the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device B

```
user@B# show interfaces
ge-1/2/0 {
   unit 0 {
      description B->D;
      family inet {
         address 172.16.1.1/24;
      }
   }
}
lo0 {
   unit 57 {
      family inet {
         address 10.0.0.1/32;
         address 10.0.0.2/32;
      }
   }
}

user@D# show protocols
bfd {
   traceoptions {
      file bfd-trace;
      flag all;
   }
}

user@B# show routing-options
static {
   route 192.168.47.0/24 {
      next-hop 172.16.1.2;
      bfd-liveness-detection {
         minimum-interval 1000;
```

```
        }
      }
    }
```

Device D

```
user@D# show interfaces
ge-1/2/0 {
  unit 1 {
    description D->B;
    family inet {
      address 172.16.1.2/24;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 192.168.47.5/32;
      address 192.168.47.6/32;
    }
  }
}

user@D# show routing-options
static {
  route 0.0.0.0/0 {
    next-hop 172.16.1.1;
    bfd-liveness-detection {
      minimum-interval 1000;
    }
  }
}
```

Verification

Confirm that the configuration is working properly.

- Verifying That BFD Sessions Are Up on page 336
- Viewing Detailed BFD Events on page 337

Verifying That BFD Sessions Are Up

Purpose Verify that the BFD sessions are up, and view details about the BFD sessions.

Action From operational mode, enter the **show bfd session extensive** command.

```
user@B> show bfd session extensive
                                                   Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
172.16.1.2               Up        lt-1/2/0.0     3.000     1.000        3
 Client Static, TX interval 1.000, RX interval 1.000
 Session up time 00:14:30
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Replicated, routing table index 172
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
```


Copyright © 2012, Juniper Networks, Inc.

```
 Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
 Local discriminator 2, remote discriminator 1
 Echo mode disabled/inactive

1 sessions, 1 clients
Cumulative transmit rate 1.0 pps, cumulative receive rate 1.0 pps

user@D> show bfd session extensive
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
172.16.1.1               Up        lt-1/2/0.1     3.000     1.000        3
 Client Static, TX interval 1.000, RX interval 1.000
 Session up time 00:14:35
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Replicated, routing table index 170
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
 Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
 Local discriminator 1, remote discriminator 2
 Echo mode disabled/inactive

1 sessions, 1 clients
Cumulative transmit rate 1.0 pps, cumulative receive rate 1.0 pps
```

Meaning The **TX interval 1.000, RX interval 1.000** output represents the setting configured with the **minimum-interval** statement. All of the other output represents the default settings for BFD. To modify the default settings, include the optional statements under the **bfd-liveness-detection** statement.

Viewing Detailed BFD Events

Purpose Check the BFD trace file to assist in troubleshooting, if needed.

Action From operational mode, enter the **file show /var/log/bfd-trace** command.

```
user@B> file show /var/log/bfd-trace
Nov 23 14:26:55    Data (9) len 35: (hex) 42 46 44 20 70 65 72 69 6f 64 69 63 20
 78 6d 69 74 20 72
Nov 23 14:26:55 PPM Trace: BFD periodic xmit rt tbl index 172
Nov 23 14:26:55 Received Downstream TraceMsg (22) len 108:
Nov 23 14:26:55    IfIndex (3) len 4: 0
Nov 23 14:26:55    Protocol (1) len 1: BFD
Nov 23 14:26:55    Data (9) len 83: (hex) 70 70 6d 64 5f 62 66 64 5f 73 65 6e 64
 6d 73 67 20 3a 20
Nov 23 14:26:55 PPM Trace: ppmd_bfd_sendmsg : socket 12 len 24, ifl 78 src
172.16.1.1 dst 172.16.1.2 errno 65
Nov 23 14:26:55 Received Downstream TraceMsg (22) len 93:
Nov 23 14:26:55    IfIndex (3) len 4: 0
Nov 23 14:26:55    Protocol (1) len 1: BFD
Nov 23 14:26:55    Data (9) len 68: (hex) 42 46 44 20 70 65 72 69 6f 64 69 63 20
 78 6d 69 74 20 74
```

Meaning BFD messages are being written to the trace file.

Related Documentation

- Understanding BFD for Static Routes on page 329

Example: Enabling BFD on Qualified Next Hops in Static Routes

This example shows how to configure a static route with multiple possible next hops. Each next hop has Bidirectional Forwarding Detection (BFD) enabled.

- Requirements on page 338
- Overview on page 338
- Configuration on page 339
- Verification on page 341

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

In this example, Device B has the static route **192.168.47.0/24** with two possible next hops. The two next hops are defined using two **qualified-next-hop** statements. Each next hop has BFD enabled.

BFD is also enabled on Device D because BFD must be enabled on both ends of the connection.

A next hop is included in the routing table if the BFD session is up. The next hop is removed from the routing table if the BFD session is down.

See Figure 51 on page 338.

Figure 51: BFD Enabled on Qualified Next Hops

Copyright © 2012, Juniper Networks, Inc.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device B

```
set interfaces fe-0/1/0 unit 2 description secondary-B->D
set interfaces fe-0/1/0 unit 2 family inet address 192.168.2.1/24
set interfaces ge-1/2/0 unit 0 description B->D
set interfaces ge-1/2/0 unit 0 family inet address 172.16.1.1/24
set routing-options static route 192.168.47.0/24 qualified-next-hop 192.168.2.2
  bfd-liveness-detection minimum-interval 60
set routing-options static route 192.168.47.0/24 qualified-next-hop 172.16.1.2
  bfd-liveness-detection minimum-interval 60
```

Device D

```
set interfaces fe-0/1/0 unit 3 description secondary-D->B
set interfaces fe-0/1/0 unit 3 family inet address 192.168.2.2/24
set interfaces ge-1/2/0 unit 1 description D->B
set interfaces ge-1/2/0 unit 1 family inet address 172.16.1.2/24
set routing-options static route 0.0.0.0/0 qualified-next-hop 192.168.2.1
set routing-options static route 0.0.0.0/0 qualified-next-hop 172.16.1.1
set routing-options static route 0.0.0.0/0 bfd-liveness-detection minimum-interval 60
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure Device B:

1. Configure the interfaces.

```
[edit interfaces fe-0/1/0]
user@B# set unit 2 description secondary-B->D
user@B# set unit 2 family inet address 192.168.2.1/24

[edit interfaces ge-1/2/0]
user@B# set unit 0 description B->D
user@B# set unit 0 family inet address 172.16.1.1/24
```

2. Configure the static route with two next hops, both with BFD enabled.

```
[edit routing-options static route 192.168.47.0/24]
user@B# set qualified-next-hop 192.168.2.2 bfd-liveness-detection minimum-interval
  60
user@B# set qualified-next-hop 172.16.1.2 bfd-liveness-detection minimum-interval
  60
```

Step-by-Step Procedure

To configure Device D:

1. Configure the interfaces:

```
[edit interfaces fe-0/1/0]
user@D# set unit 3 description secondary-D->B
user@D# set unit 3 family inet address 192.168.2.2/24
```

```
[edit interfaces ge-1/2/0]
user@D# set unit 1 description D->B
user@D# set unit 1 family inet address 172.16.1.2/24
```

2. Configure a BFD-enabled default static route with two next hops to the provider network.

   In this case, BFD is enabled on the route, not on the next hops.

```
[edit routing-options static route 0.0.0.0/0]
user@D# set qualified-next-hop 192.168.2.1
user@D# set qualified-next-hop 172.16.1.1
user@D# set bfd-liveness-detection minimum-interval 60
```

Results Confirm your configuration by issuing the **show interfaces** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@B# show interfaces
fe-0/1/0 {
   unit 2 {
      description secondary-B->D;
      family inet {
         address 192.168.2.1/24;
      }
   }
}
ge-1/2/0 {
   unit 0 {
      description B->D;
      family inet {
         address 172.16.1.1/24;
      }
   }
}
```

```
user@B# show routing-options
static {
   route 192.168.47.0/24 {
      qualified-next-hop 192.168.2.2 {
         bfd-liveness-detection {
            minimum-interval 60;
         }
      }
      qualified-next-hop 172.16.1.2 {
         bfd-liveness-detection {
            minimum-interval 60;
         }
      }
   }
}
```

```
user@D# show interfaces
fe-0/1/0 {
   unit 3 {
      description secondary-D->B;
      family inet {
```

 Copyright © 2012, Juniper Networks, Inc.

```
            address 192.168.2.2/24;
          }
        }
      }
      ge-1/2/0 {
        unit 1 {
          description D->B;
          family inet {
            address 172.16.1.2/24;
          }
        }
      }

      user@D# show routing-options
      static {
        route 0.0.0.0/0 {
          qualified-next-hop 192.168.2.1;
          qualified-next-hop 172.16.1.1;
          bfd-liveness-detection {
            minimum-interval 60;
          }
        }
      }
```

If you are done configuring the devices, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Checking the Routing Tables on page 341
- Verifying the BFD Sessions on page 342
- Removing BFD from Device D on page 342
- Removing BFD from One Next Hop on page 343

Checking the Routing Tables

Purpose Make sure that the static route appears in the routing table on Device B with two possible next hops.

Action

```
user@B> show route 192.168.47.0 extensive
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
192.168.47.0/24 (1 entry, 1 announced)
TSI:
KRT in-kernel 192.168.47.0/24 -> {192.168.2.2}
        *Static Preference: 5
                Next hop type: Router
                Address: 0x9334010
                Next-hop reference count: 1
                Next hop: 172.16.1.2 via ge-1/2/0.0
                Next hop: 192.168.2.2 via fe-0/1/0.2, selected
                State: <Active Int Ext>
                Age: 9
                Task: RT
```

```
                    Announcement bits (1): 3-KRT
                    AS path: I
```

Meaning Both next hops are listed. The next hop 192.168.2.2 is the selected route.

Verifying the BFD Sessions

Purpose Make sure that the BFD sessions are up.

Action

```
user@B> show bfd session

                                          Detect   Transmit
Address          State     Interface      Time     Interval  Multiplier
172.16.1.2       Up        ge-1/2/0.0     0.720     0.240        3
192.168.2.2      Up        fe-0/1/0.2     0.720     0.240        3

2 sessions, 2 clients
Cumulative transmit rate 8.3 pps, cumulative receive rate 8.3 pps
```

Meaning The output shows that the BFD sessions are up.

Removing BFD from Device D

Purpose Demonstrate what happens when the BFD session is down for both next hops.

Action

1. Deactivate BFD on Device D.

   ```
   [edit routing-options static route 0.0.0.0/0]
   user@D# deactivate bfd-liveness-detection
   user@D# commit
   ```

2. Rerun the **show bfd session** command on Device B.

   ```
   user@B> show bfd session


                                                        Detect   Transmit
   Address                  State     Interface      Time     Interval
   Multiplier
   172.16.1.2               Down      ge-1/2/0.0     3.000     1.000
    3
   192.168.2.2              Down      fe-0/1/0.2     3.000     1.000
    3

   2 sessions, 2 clients
   Cumulative transmit rate 2.0 pps, cumulative receive rate 2.0 pps
   ```

3. Rerun the **show route 192.168.47.0** command on Device B.

   ```
   user@B> show route 192.168.47.0
   ```

Meaning As expected, when the BFD sessions are down, the static route is removed from the routing table.

 Copyright © 2012, Juniper Networks, Inc.

Removing BFD from One Next Hop

Purpose Demonstrate what happens when only one next hop has BFD enabled.

Action

1. If it is not already deactivated, deactivate BFD on Device D.

   [edit routing-options static route 0.0.0.0/0]
   user@D# **deactivate bfd-liveness-detection**
   user@D# **commit**

2. Deactivate BFD on one of the next hops on Device B.

   [edit routing-options static route 192.168.47.0/24 qualified-next-hop 172.16.1.2]
   user@B# **deactivate bfd-liveness-detection**
   user@B# **commit**

3. Rerun the **show bfd session** command on Device B.

```
user@B> show bfd session


                                        Detect   Transmit
Address       State     Interface       Time     Interval  Multiplier
192.168.2.2    Down      fe-0/1/0.2      3.000     1.000        3
```

4. Rerun the **show route 192.168.47.0 extensive** command on Device B.

```
user@B> show route 192.168.47.0 extensive


inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
192.168.47.0/24 (1 entry, 1 announced)
TSI:
KRT in-kernel 192.168.47.0/24 -> {172.16.1.2}
        *Static Preference: 5
                Next hop type: Router, Next hop index: 624
                Address: 0x92f0178
                Next-hop reference count: 3
                Next hop: 172.16.1.2 via ge-1/2/0.0, selected
                State: <Active Int Ext>
                Age: 2:36
                Task: RT
                Announcement bits (1): 3-KRT
                AS path: I
```

Meaning As expected, the BFD session is down for the 192.168.2.2 next hop. The 172.16.1.2 next hop remains in the routing table, and the route remains active, because BFD is not a condition for this next hop to remain valid.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring Static Route Preferences and Qualified Next Hops
- Understanding BFD for Static Routes on page 329
- Verifying the Static Route Configuration on page 89

BFD Authentication and Static Routes

- Understanding BFD Authentication for Static Routes on page 344
- Example: Configuring BFD Authentication for Static Routes on page 346

Understanding BFD Authentication for Static Routes

Bidirectional Forwarding Detection (BFD) enables rapid detection of communication failures between adjacent systems. By default, authentication for BFD sessions is disabled. However, when you run BFD over Network Layer protocols, the risk of service attacks can be significant.

NOTE: **We strongly recommend using authentication if you are running BFD over multiple hops or through insecure tunnels.**

Beginning with Junos OS Release 9.6, Junos OS supports authentication for BFD sessions running over IPv4 and IPv6 static routes. BFD authentication is not supported on MPLS OAM sessions. BFD authentication is only supported in the domestic image and is not available in the export image.

You authenticate BFD sessions by specifying an authentication algorithm and keychain, and then associating that configuration information with a security authentication keychain using the keychain name.

The following sections describe the supported authentication algorithms, security keychains, and level of authentication that can be configured:

- BFD Authentication Algorithms on page 344
- Security Authentication Keychains on page 345
- Strict Versus Loose Authentication on page 345

BFD Authentication Algorithms

Junos OS supports the following algorithms for BFD authentication:

- **simple-password**—Plain-text password. One to 16 bytes of plain text are used to authenticate the BFD session. One or more passwords can be configured. This method is the least secure and should be used only when BFD sessions are not subject to packet interception.
- **keyed-md5**—Keyed Message Digest 5 hash algorithm for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed MD5 uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than or equal to the last sequence number received. Although more secure than a simple password, this method is vulnerable to replay attacks. Increasing the rate at which the sequence number is updated can reduce this risk.

Copyright © 2012, Juniper Networks, Inc.

- **meticulous-keyed-md5**—Meticulous keyed Message Digest 5 hash algorithm. This method works in the same manner as keyed MD5, but the sequence number is updated with every packet. Although more secure than keyed MD5 and simple passwords, this method might take additional time to authenticate the session.
- **keyed-sha-1**—Keyed Secure Hash Algorithm I for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed SHA uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. The key is not carried within the packets. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than the last sequence number received.
- **meticulous-keyed-sha-1**—Meticulous keyed Secure Hash Algorithm I. This method works in the same manner as keyed SHA, but the sequence number is updated with every packet. Although more secure than keyed SHA and simple passwords, this method might take additional time to authenticate the session.

NOTE: **Nonstop active routing (NSR) is not supported with meticulous-keyed-md5 and meticulous-keyed-sha-1 authentication algorithms. BFD sessions using these algorithms might go down after a switchover.**

Security Authentication Keychains

The security authentication keychain defines the authentication attributes used for authentication key updates. When the security authentication keychain is configured and associated with a protocol through the keychain name, authentication key updates can occur without interrupting routing and signaling protocols.

The authentication keychain contains one or more keychains. Each keychain contains one or more keys. Each key holds the secret data and the time at which the key becomes valid. The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

BFD allows multiple clients per session, and each client can have its own keychain and algorithm defined. To avoid confusion, we recommend specifying only one security authentication keychain.

Strict Versus Loose Authentication

By default, strict authentication is enabled and authentication is checked at both ends of each BFD session. Optionally, to smooth migration from nonauthenticated sessions to authenticated sessions, you can configure *loose checking*. When loose checking is configured, packets are accepted without authentication being checked at each end of the session. This feature is intended for transitional periods only.

Related Documentation

- Example: Configuring BFD Authentication for Static Routes on page 346

Example: Configuring BFD Authentication for Static Routes

This example shows how to configure bidirectional forwarding detection (BFD) authentication for static routes.

- Requirements on page 346
- Overview on page 346
- Configuration on page 347
- Verification on page 350

Requirements

Junos OS Release 9.6 or later (domestic).

BFD authentication is only supported in the domestic image of Junos OS and is not available in the export image.

Overview

You can configure authentication for BFD sessions running over IPv4 and IPv6 static routes. Routing instances and logical systems are also supported.

The following steps are needed to configure authentication on a BFD session:

1. Specify the BFD authentication algorithm for the static route.
2. Associate the authentication keychain with the static route.
3. Configure the related security authentication keychain. This must be configured on the main router.

TIP: We recommend that you specify loose authentication checking if you are transitioning from nonauthenticated sessions to authenticated sessions.

```
[edit]
user@host> set routing-options static route ipv4 bfd-liveness-detection
  authentication loose-check
```

Figure 52 on page 347 shows the sample network.

Copyright © 2012, Juniper Networks, Inc.

Figure 52: Customer Routes Connected to a Service Provider

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device B

```
set interfaces ge-1/2/0 unit 0 description B->D
set interfaces ge-1/2/0 unit 0 family inet address 172.16.1.1/24
set interfaces lo0 unit 57 family inet address 10.0.0.1/32
set interfaces lo0 unit 57 family inet address 10.0.0.2/32
set routing-options static route 192.168.47.0/24 next-hop 172.16.1.2
set routing-options static route 192.168.47.0/24 bfd-liveness-detection minimum-interval
    1000
set routing-options static route 192.168.47.0/24 bfd-liveness-detection authentication
    key-chain bfd-kc4
set routing-options static route 192.168.47.0/24 bfd-liveness-detection authentication
    algorithm keyed-sha-1
set security authentication-key-chains key-chain bfd-kc4 key 5 secret
    "$9$JhZHmn6Ap0Ir/9ApOcSs24oaZIkPfT3wY24ZGmz36AtOIEyMWxSrlKvM-dbs2aDkP5Ft0IQFdev7N"
set security authentication-key-chains key-chain bfd-kc4 key 5 start-time
    "2011-1-1.12:00:00 -0800"
```

Device D

```
set interfaces ge-1/2/0 unit 1 description D->B
set interfaces ge-1/2/0 unit 1 family inet address 172.16.1.2/24
set interfaces lo0 unit 2 family inet address 192.168.47.5/32
set interfaces lo0 unit 2 family inet address 192.168.47.6/32
set routing-options static route 0.0.0.0/0 next-hop 172.16.1.1
set routing-options static route 0.0.0.0/0 bfd-liveness-detection minimum-interval 1000
set routing-options static route 0.0.0.0/0 bfd-liveness-detection authentication key-chain
    bfd-kc4
set routing-options static route 0.0.0.0/0 bfd-liveness-detection authentication algorithm
    keyed-sha-1
```

```
set security authentication-key-chains key-chain bfd-kc4 key 5 secret
   "$9$JhZHmn6Ap0In/9ApOcSs24oaZikPfT3wY24ZGmz36AtOIEyMWxSrlKvM-dbs2aDkP5Ft0IQFdev7N"
set security authentication-key-chains key-chain bfd-kc4 key 5 start-time
   "2011-1-1.12:00:00 -0800"
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure BFD for static routes:

1. On Device B, configure the interfaces.

```
[edit interfaces]
user@B# set ge-1/2/0 unit 0 description B->D
user@B# set ge-1/2/0 unit 0 family inet address 172.16.1.1/24

user@B# set lo0 unit 57 family inet address 10.0.0.1/32
user@B# set lo0 unit 57 family inet address 10.0.0.2/32
```

2. On Device B, create a static route and set the next-hop address.

```
[edit routing-options]
user@B# set static route 192.168.47.0/24 next-hop 172.16.1.2
```

3. On Device B, configure BFD for the static route.

```
[edit routing-options]
user@B# set static route 192.168.47.0/24 bfd-liveness-detection minimum-interval
   1000
```

4. On Device B, configure the authentication keychain.

   Specify the unique security authentication information for BFD sessions:

   - The matching *key-chain-name*.
   - At least one *key*, a unique integer between **0** and **63**. Creating multiple keys allows multiple clients to use the BFD session.
   - The *secret-data* used to allow access to the session.
   - The time at which the authentication key becomes active, *yyyy-mm-dd.hh:mm:ss*.

```
[edit security authentication-key-chains key-chain bfd-kc4]
user@B# set key 5 secret
   "$9$JhZHmn6Ap0In/9ApOcSs24oaZikPfT3wY24ZGmz36AtOIEyMWxSrlKvM-dbs2aDkP5Ft0IQFdev7N"
user@B# set key 5 start-time "2011-1-1.12:00:00 -0800"
```

5. On Device B, specify the keychain to be used to associate BFD sessions on the specified route with the unique security authentication keychain attributes.

   This should match the keychain name configured at the **[edit security authentication key-chains]** hierarchy level.

```
[edit routing-options]
user@B# set static route 192.168.47.0/24 bfd-liveness-detection authentication
   key-chain bfd-kc4
```


Copyright © 2012, Juniper Networks, Inc.

6. On Device B, specify the algorithm (**keyed-md5**, **keyed-sha-1**, **meticulous-keyed-md5**, **meticulous-keyed-sha-1**, or **simple-password**) to use for BFD authentication on the static route.

```
[edit routing-options]
user@B# set static route 192.168.47.0/24 bfd-liveness-detection authentication
  algorithm keyed-sha-1
```

NOTE: Nonstop active routing (NSR) is not supported with meticulous-keyed-md5 and meticulous-keyed-sha-1 authentication algorithms. BFD sessions using these algorithms might go down after a switchover.

7. If you are done configuring Device B, commit the configuration.

```
[edit]
user@B# commit
```

8. Repeat the configuration on Device D.

   The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

Results Confirm your configuration by issuing the **show interfaces**, **show routing-options**, and **show security** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device B

```
user@B# show interfaces
ge-1/2/0 {
   unit 0 {
      description B->D;
      family inet {
         address 172.16.1.1/24;
      }
   }
}
lo0 {
   unit 57 {
      family inet {
         address 10.0.0.1/32;
         address 10.0.0.2/32;
      }
   }
}

user@B# show routing-options
static {
   route 192.168.47.0/24 {
      next-hop 172.16.1.2;
      bfd-liveness-detection {
         minimum-interval 1000;
         authentication {
```

```
            key-chain bfd-kc4;
            algorithm keyed-sha-1;
          }
        }
      }
    }

    user@B# show security
    authentication-key-chains {
      key-chain bfd-kc4 {
        key 5 {
          secret
            "$9$JhZHmn6Ap0In/9ApOcSs24oaZ1kPfT3wY24ZGmz36AtOIEyMWxSrIKvM-dbs2aDkP5Ft0IQFdev7N";
            ## SECRET-DATA
          start-time "2011-1-1.12:00:00 -0800";
        }
      }
    }
```

Verification

Confirm that the configuration is working properly.

- Verifying That BFD Sessions Are Up on page 350
- Viewing Details About the BFD Session on page 350
- Viewing Extensive BFD Session Information on page 351

Verifying That BFD Sessions Are Up

Purpose Verify that the BFD sessions are up.

Action From operational mode, enter the **show bfd session** command.

```
user@B> show bfd session
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
172.16.1.2               Up        ge-1/2/0.0     3.000     1.000        3

1 sessions, 1 clients
Cumulative transmit rate 1.0 pps, cumulative receive rate 1.0 pps
```

Meaning The command output shows that the BFD session is up.

Viewing Details About the BFD Session

Purpose View details about the BFD sessions and make sure that authentication is configured.

Action From operational mode, enter the **show bfd session detail** command.

```
user@B> show bfd session detail
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
172.16.1.2               Up        ge-1/2/0.0     3.000     1.000        3
 Client Static, TX interval 1.000, RX interval 1.000, Authenticate
 Session up time 00:53:58
 Local diagnostic NbrSignal, remote diagnostic None
```

 Copyright © 2012, Juniper Networks, Inc.

```
 Remote state Up, version 1
 Logical system 9, routing table index 22

1 sessions, 1 clients
Cumulative transmit rate 1.0 pps, cumulative receive rate 1.0 pps
```

Meaning In the command output, **Authenticate** is displayed to indicate that BFD authentication is configured.

Viewing Extensive BFD Session Information

Purpose View more detailed information about the BFD sessions.

Action From operational mode, enter the **show bfd session extensive** command.

```
user@B> show bfd session extensive
Address                  State     Interface      Time     Interval  Multiplier
172.16.1.2               Up        ge-1/2/0.0     3.000     1.000        3
 Client Static, TX interval 1.000, RX interval 1.000, Authenticate
        keychain bfd-kc4, algo keyed-sha-1, mode strict
 Session up time 01:39:45
 Local diagnostic NbrSignal, remote diagnostic None
 Remote state Up, version 1
 Logical system 9, routing table index 22
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
 Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
 Local discriminator 3, remote discriminator 4
 Echo mode disabled/inactive
 Authentication enabled/active, keychain bfd-kc4, algo keyed-sha-1, mode strict

1 sessions, 1 clients
Cumulative transmit rate 1.0 pps, cumulative receive rate 1.0 pps
```

Meaning In the command output, **Authenticate** is displayed to indicate that BFD authentication is configured. The output for the **show bfd sessions extensive** command provides the keychain name, the authentication algorithm, and the mode for each client in the session.

Related Documentation

- Understanding BFD Authentication for Static Routes on page 344

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 13

Multicast

- Multicast Overview on page 353
- Multicast Configuration Overview on page 359
- SAP and SDP Multicast Session Announcements on page 360
- Multicast IGMP on page 362
- Multicast PIM and Static RPs on page 367
- PIM Register Messages on page 370
- PIM RPF Routing Tables on page 377
- Verifying a Multicast Configuration on page 382

Multicast Overview

NOTE: **Both Protocol Independent Multicast (PIM) version 1 and PIM version 2 are supported. In this topic, the term *PIM* refers to both versions of the protocol.**

Multicast traffic lies between the extremes of unicast (one source, one destination) and broadcast (one source, all destinations). Multicast is a “one source, many destinations” method of traffic distribution, meaning that the destinations needing to receive the information from a particular source receive the traffic stream.

IP network destinations (clients) do not often communicate directly with sources (servers), so the routers between source and destination must be able to determine the topology of the network from the unicast or multicast perspective to avoid routing traffic haphazardly. The multicast router must find multicast sources on the network, send out copies of packets on several interfaces, prevent routing loops, connect interested destinations with the proper source, and keep the flow of unwanted packets to a minimum. Standard multicast routing protocols provide most of these capabilities.

This chapter contains the following topics:

- Multicast Architecture on page 354
- Dense and Sparse Routing Modes on page 355

- Strategies for Preventing Routing Loops on page 356
- Multicast Protocol Building Blocks on page 357

Multicast Architecture

Multicast-capable routers replicate packets on the multicast network, which has exactly the same topology as the unicast network it is based on. Multicast routers use a multicast routing protocol to build a distribution tree that connects receivers (also called *listeners*) to sources.

Upstream and Downstream Interfaces

A single upstream interface on the router leads toward the source to receive multicast packets. The downstream interfaces on the router lead toward the receivers to transmit packets. A router can have as many downstream interfaces as it has logical interfaces, minus 1. To prevent looping, the router's upstream interface must never receive copies of its own downstream multicast packets.

Subnetwork Leaves and Branches

On a multicast router, each subnetwork of hosts that includes at least one interested receiver is a leaf on the multicast distribution tree (see Figure 53 on page 355). The router must send out a copy of the IP multicast packet on each interface with a leaf. When a new leaf subnetwork joins the tree, a new branch is built so that the router can send out replicated packets on the interface. The number of leaves on an interface does not affect the router. The action is the same for one leaf or a hundred.

NOTE: On Juniper Networks security devices, if the maximum number of leaves on a multicast distribution tree is exceeded, multicast sessions are created up to the maximum number of leaves, and any multicast sessions that exceed the maximum number of leaves are ignored. The maximum number of leaves on a multicast distribution tree is device specific.

A branch that no longer has leaves is pruned from the distribution tree. No multicast packets are sent out on a router interface leading to an IP subnetwork with no interested hosts. Because packets are replicated only where the distribution tree branches, no link ever carries a duplicate flow of packets.

In IP multicast networks, traffic is delivered to multicast groups based on an IP multicast group address instead of a unicast destination address. The groups determine the location of the leaves, and the leaves determine the branches on the multicast network.

Copyright © 2012, Juniper Networks, Inc.

Figure 53: Multicast Elements in an IP Network

Multicast IP Address Ranges

Multicast uses the Class D IP address range (**224.0.0.0** through **239.255.255.255**). Multicast addresses usually have a prefix length of **/32**, although other prefix lengths are allowed. Multicast addresses represent logical groupings of receivers and not physical collections of devices, and can appear only as the destination in an IP packet, never as the source address.

Notation for Multicast Forwarding States

The multicast forwarding state in a router is usually represented by one of the following notations:

- (S,G) notation—S refers to the unicast IP address of the source for the multicast traffic and G refers to the particular multicast group IP address for which S is the source. All multicast packets sent from this source have S as the source address and G as the destination address.
- (*, G) notation—The asterisk (*) is a wildcard for the address of any multicast application source sending to group G. For example, if two sources are originating exactly the same content for multicast group **224.1.1.2**, a router can use (*, **224.1.1.2**) to represent the state of a router forwarding traffic from both sources to the group.

Dense and Sparse Routing Modes

To keep packet replication to a minimum, multicast routing protocols use the two primary modes shown in Table 8 on page 356.

CAUTION: A common multicast guideline is not to run dense mode on a WAN under any circumstances.

Table 8: Primary Multicast Routing Modes

Multicast Mode	Description	Appropriate Network for Use
Dense mode	Network is flooded with traffic on all possible branches, then pruned back as branches explicitly (by message) or implicitly (time-out silence) eliminate themselves.	LANs—Networks in which all possible subnets are likely to have at least one receiver.
Sparse mode	Network establishes and sends packets only on branches that have at least one leaf indicating (by message) a need for the traffic.	WANs—Network in which very few of the possible receivers require packets from this source.

Strategies for Preventing Routing Loops

Routing loops are disastrous in multicast networks because of the risk of repeatedly replicated packets, which can overwhelm a network. One of the complexities of modern multicast routing protocols is the need to avoid routing loops, packet by packet, much more rigorously than in unicast routing protocols. Three multicast strategies—reverse-path forwarding (RPF), shortest-path tree (SPT), and administrative scoping—help prevent routing loops by defining routing paths in different ways.

Reverse-Path Forwarding for Loop Prevention

The router's multicast forwarding state runs more logically based on the reverse path, from the receiver back to the root of the distribution tree. In RPF, every multicast packet received must pass an RPF check before it can be replicated or forwarded on any interface. When it receives a multicast packet on an interface, the router verifies that the *source* address in the multicast IP packet is the *destination* address for a unicast IP packet back to the source.

If the outgoing interface found in the unicast routing table is the same interface that the multicast packet was received on, the packet passes the RPF check. Multicast packets that fail the RPF check are dropped, because the incoming interface is not on the shortest path back to the source. Routers can build and maintain separate tables for RPF purposes. See "Understanding PIM RPF Routing Tables" on page 378.

Shortest-Path Tree for Loop Prevention

The distribution tree used for multicast is rooted at the source and is the shortest-path tree (SPT), but this path can be long if the source is at the periphery of the network. Providing a *shared tree* on the backbone as the distribution tree locates the multicast source more centrally in the network. Shared distribution trees with roots in the core network are created and maintained by a multicast router operating as a rendezvous point (RP), a feature of sparse mode multicast protocols.

Administrative Scoping for Loop Prevention

Scoping limits the routers and interfaces that can forward a multicast packet. Multicast scoping is *administrative* in the sense that a range of multicast addresses is reserved for scoping purposes, as described in RFC 2365, *Administratively Scoped IP Multicast*. Routers at the boundary must filter multicast packets and ensure that packets do not stray beyond the established limit.

 Copyright © 2012, Juniper Networks, Inc.

Multicast Protocol Building Blocks

Multicast is not a single protocol, but a collection of protocols working together to form trees, prune branches, locate sources and groups, and prevent routing loops:

- Distance Vector Multicast Routing Protocol (DVMRP) and Protocol Independent Multicast (PIM) operate between routers. PIM can operate in dense mode and sparse mode.
- Three versions of the Internet Group Management Protocol (IGMP) run between receiver hosts and routers.
- Several other routing mechanisms and protocols enhance multicast networks by providing useful functions not included in other protocols. These include the bootstrap router (BSR) mechanism, auto-rendezvous point (RP), Multicast Source Discovery Protocol (MSDP), Session Announcement Protocol (SAP), Session Description Protocol (SDP), and Pragmatic General Multicast (PGM) protocol.

Table 9 on page 357 lists and summarizes these protocols.

Table 9: Multicast Protocol Building Blocks

Multicast Protocol	Description	Uses
DVMRP	Dense-mode-only protocol that uses the flood-and-prune or implicit join method to deliver traffic everywhere and then determine where the uninterested receivers are. DVRMP uses source-based distribution trees in the form (S,G) and builds its own multicast routing tables for RPF checks.	Not appropriate for large-scale Internet use.
PIM dense mode	Sends an *implicit* join message, so routers use the flood-and-prune method to deliver traffic everywhere and then determine where the uninterested receivers are. PIM dense mode uses source-based distribution trees in the form (S,G), and also supports sparse-dense mode, with mixed sparse and dense groups. Both PIM modes use unicast routing information for RPF checks.	Most promising multicast protocol in use for LANs.

Copyright © 2012, Juniper Networks, Inc.

Table 9: Multicast Protocol Building Blocks *(continued)*

Multicast Protocol	Description	Uses
PIM sparse "Understanding IGMP and Multicast" on page 362mode	Sends an *explicit* join message, so routers determine where the interested receivers are and send join messages upstream to their neighbors, building trees from receivers to an RP router, which is the initial source of multicast group traffic. PIM sparse mode builds distribution trees in the form (*,G), but migrates to an (S,G) source-based tree if that path is shorter than the path through the RP router for a particular multicast group's traffic. Both PIM modes use unicast routing information for RPF checks.	Most promising multicast protocol in use for WANs. See "Understanding PIM and Static RPs" on page 367.
PIM source-specific multicast (SSM)	Enhancement to PIM sparse mode that allows a client to receive multicast traffic directly from the source, without the help of an RP.	Used with IGMPv3 to create a shortest-path tree between receiver and source.
IGMPv1	The original protocol defined in RFC 1112, *Host Extensions for IP Multicasting*. IGMPv1 sends an explicit join message to the router, but uses a timeout to determine when hosts leave a group.	See "Understanding IGMP and Multicast" on page 362.
IGMPv2	Defined in RFC 2236, *Internet Group Management Protocol, Version 2*. Among other features, IGMPv2 adds an explicit leave message to the join message.	Used by default. ee "Understanding IGMP and Multicast" on page 362.
IGMPv3	Defined in RFC 3376, *Internet Group Management Protocol, Version 3.* Among other features, IGMPv3 optimizes support for a single source of content for a multicast group, or source-specific multicast (SSM).	Used with PIM SSM to create a shortest-path tree between receiver and source.
BSR and Auto-RP	Allow sparse-mode routing protocols to find RPs within the routing domain (autonomous system, or AS). RP addresses can also be statically configured.	
MSDP "Understanding SAP and SDP Multicast Session Announcements" on page 360.	Allows groups located in one multicast routing domain to find RPs in other routing domains. MSDP is not used on an RP if all receivers and sources are located in the same routing domain.	Typically runs on the same router as PIM sparse mode RP. Not appropriate if all receivers and sources are located in the same routing domain.

Copyright © 2012, Juniper Networks, Inc.

Table 9: Multicast Protocol Building Blocks *(continued)*

Multicast Protocol	Description	Uses
SAP and SDP	Display multicast session names and correlate the names with multicast traffic. SDP is a session directory protocol that advertises multimedia conference sessions and communicates setup information to participants who want to join the session. A client commonly uses SDP to announce a conference session by periodically multicasting an announcement packet to a well-known multicast address and port using SAP.	
PGM	Special protocol layer for multicast traffic that can be used between the IP layer and the multicast application to add reliability to multicast traffic. PGM allows a receiver to detect missing information in all cases and request replacement information if the receiver application requires it.	

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Overview on page 3
- Multicast Configuration Overview on page 359
- Multicast Overview in the *Junos OS Multicast Protocols Configuration Guide*

Multicast Configuration Overview

You configure a router network to support multicast applications with a related family of protocols. To use multicast, you must understand the basic components of a multicast network and their relationships, and then configure the device to act as a node in the network.

To configure the device as a node in a multicast network:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to use the sparse, dense, or sparse-dense mode of multicast operation. Each mode has different configuration considerations.
4. Determine the address of the rendezvous point (RP) if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, bootstrap router (BSR), or auto-RP method.

Copyright © 2012, Juniper Networks, Inc.

6. Determine whether to configure multicast to use its own reverse-path forwarding (RPF) routing table when configuring PIM in sparse, dense, or sparse-dense modes.
7. (Optional) Configure the SAP and SDP protocols to listen for multicast session announcements. See "Example: Configuring SAP and SDP to Listen for Session Announcements" on page 361.
8. Configure IGMP. See "Example: Configuring IGMP for Multicast" on page 362.
9. (Optional) Configure the PIM static RP. See "Understanding PIM and Static RPs" on page 367.
10. (Optional) Filter PIM register messages from unauthorized groups and sources. See "Example: Rejecting Incoming PIM Register Messages on RP Routers" on page 371 and "Example: Stopping Outgoing PIM Register Messages on a Designated Router" on page 374.
11. (Optional) Configure a PIM RPF routing table. See "Example: Configuring a PIM RPF Routing Table" on page 378.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Multicast Overview on page 353
- Verifying a Multicast Configuration on page 382

SAP and SDP Multicast Session Announcements

- Understanding SAP and SDP Multicast Session Announcements on page 360
- Example: Configuring SAP and SDP to Listen for Session Announcements on page 361

Understanding SAP and SDP Multicast Session Announcements

Multicast session announcements are handled by two protocols, the Session Announcement Protocol (SAP), and Session Description Protocol (SDP). These two protocols display multicast session names and correlate the names with multicast traffic. Enabling SDP and SAP allows the router to receive announcements about multimedia and other multicast sessions from sources. Enabling SAP automatically enables SDP.

The device listens for session announcements on one or more addresses and ports. By default, the router listens to address and port **224.2.127.254:9875**.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Multicast Overview on page 353
- Example: Configuring SAP and SDP to Listen for Session Announcements on page 361

 Copyright © 2012, Juniper Networks, Inc.

Example: Configuring SAP and SDP to Listen for Session Announcements

This example shows how to configure SAP and SDP.

- Requirements on page 361
- Overview on page 361
- Configuration on page 361
- Verification on page 362

Requirements

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its own RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.

Overview

In this example, you set the address value to the IP address as **224.2.127.254** and set the port value to **9875**.

Configuration

Step-by-Step Procedure

To configure SAP and SDP:

1. Configure SAP.

```
[edit]
user@host# edit protocols sap
```

2. Set the address value and the port value.

```
[edit]
user@host# set listen 224.2.127.254 port 9875
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show protocols sap listen** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding SAP and SDP Multicast Session Announcements on page 360
- Multicast Configuration Overview on page 359
- Verifying a Multicast Configuration on page 382

Multicast IGMP

- Understanding IGMP and Multicast on page 362
- Example: Configuring IGMP for Multicast on page 362
- IPv6 Multicast Flow on page 364

Understanding IGMP and Multicast

The Internet Group Management Protocol (IGMP) manages the membership of hosts and routers in multicast groups. IGMP is an integral part of IP and must be enabled on all routers and hosts that need to receive IP multicasts. IGMP is automatically enabled on all broadcast interfaces when you configure PIM or DVMRP.

By default, the device runs IGMPv2. However, you might still want to set the IGMP version explicitly on an interface, or all interfaces. Routers running different versions of IGMP negotiate the lowest common version of IGMP supported by hosts on their subnet. One host running IGMPv1 forces the device to use that version and lose features important to other hosts.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Multicast Overview on page 353
- Example: Configuring IGMP for Multicast on page 362
- Understanding IGMP in the *Junos OS Multicast Protocols Configuration Guide*

Example: Configuring IGMP for Multicast

This example shows how to configure IGMP for multicast.

- Requirements on page 363
- Overview on page 363
- Configuration on page 363
- Verification on page 363

Copyright © 2012, Juniper Networks, Inc.

Requirements

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its own RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.
7. Configure the SAP and SDP protocols to listen for multicast session announcements. See "Example: Configuring SAP and SDP to Listen for Session Announcements" on page 361.

Overview

In this example, you set the interface value to **all** and the version number to **2**.

Configuration

Step-by-Step Procedure

To configure IGMP for multicast:

1. Configure the interface level.

   ```
   [edit]
   user@host# edit protocols igmp
   ```

2. Set the interface value and the version number.

   ```
   [edit]
   user@host# set igmp interface all version 2
   ```

3. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Verification

To verify the configuration is working properly, enter the **show igmp interface** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding IGMP and Multicast on page 362
- Multicast Configuration Overview on page 359

Copyright © 2012, Juniper Networks, Inc.

- Configuring IGMP in the *Junos OS Multicast Protocols Configuration Guide*
- Enabling IGMP in the *Junos OS Multicast Protocols Configuration Guide*
- Verifying a Multicast Configuration on page 382

IPv6 Multicast Flow

- IPv6 Multicast Flow Overview on page 364
- Multicast Listener Discovery (MLD) Overview on page 365

IPv6 Multicast Flow Overview

The IPv6 multicast flow adds or enhances the following features:

- IPv6 transit multicast which includes the following packet functions:
  - Normal packet handling
  - Fragment handling
  - Packet reordering
- Protocol-Independent Multicast version 6 (PIMv6) flow handling
- Other multicast routing protocols, such as Multicast Listener Discovery (MLD)

The structure and processing of IPv6 multicast data session are the same as those of IPv4. Each data session has the following:

- One template session
- Several leaf sessions.

The reverse path forwarding (RPF) check behavior for IPv6 is the same as that for IPv4. Incoming multicast data is accepted only if the RPF check succeeds. In an IPv6 multicast flow, incoming Multicast Listener Discovery (MLD) protocol packets are accepted only if MLD or PIM is enabled in the security zone for the incoming interface. Sessions for multicast protocol packets have a default timeout value of 300 seconds. This value cannot be configured. The null register packet is sent to rendezvous point (RP).

In IPv6 multicast flow, a multicast router has the following three roles:

- Designated router

  This router receives the multicast packets, encapsulates them with unicast IP headers, and sends them for multicast flow.

- Intermediate router

  There are two sessions for the packets, the control session, for the outer unicast packets, and the data session. The security policies are applied to the data session and the control session, is used for forwarding.

- Rendezvous point

 Copyright © 2012, Juniper Networks, Inc.

The RP receives the unicast PIM register packet, separates the unicast header, and then forwards the inner multicast packet. The packets received by RP are sent to the pd interface for decapsulation and are later handled like normal multicast packets.

On a Services Processing Unit (SPU), the multicast session is created as a template session for matching the incoming packet's tuple. Leaf sessions are connected to the template session. On the Customer Premise Equipment (CPE), only the template session is created. Each CPE session carries the fan-out lists that are used for load-balanced distribution of multicast SPU sessions.

NOTE: **IPv6 multicast uses the IPv4 multicast behavior for session distribution.**

The network service access point identifier (nsapi) of the leaf session is set up on the multicast text traffic going into the tunnels, to point to the outgoing tunnel. The zone ID of the tunnel is used for policy lookup for the leaf session in the second stage. Multicast packets are unidirectional. Thus for multicast text session sent into the tunnels, forwarding sessions are not created.

When the multicast route ages out, the corresponding chain of multicast sessions is deleted. When the multicast route changes, then the corresponding chain of multicast sessions is deleted. This forces the next packet hitting the multicast route to take the first path and re-create the chain of sessions; the multicast route counter is not affected.

NOTE: **The IPv6 multicast packet reorder approach is same as that for IPv4.**

For the encapsulating router, the incoming packet is multicast, and the outgoing packet is unicast. For the intermediate router, the incoming packet is unicast, and the outgoing packet is unicast.

Multicast Listener Discovery (MLD) Overview

The Multicast Listener Discovery (MLD) protocol manages the membership of hosts and routers in multicast groups. IP version 6 (IPv6) multicast routers use MLD to learn, for each of their attached physical networks, which groups have interested listeners. Each router maintains a list of host multicast addresses that have listeners for each subnetwork, as well as a timer for each address. However, the router does not need to know the address of the listeners—just the address of the hosts. The router provides addresses to the multicast routing protocol it uses; this ensures that multicast packets are delivered to all subnetworks where there are interested listeners. In this way, MLD is used as the transport for the Protocol Independent Multicast (PIM) protocol.

MLD is an integral part of IPv6 and must be enabled on all IPv6 routers and hosts that need to receive IP multicast traffic. Junos OS supports MLD versions 1 and 2. Version 2 is supported for the source-specific multicast (SSM) include and exclude modes.

In include mode, the receiver specifies the source or sources from which it is interested in receiving the multicast group traffic. Exclude mode works the opposite of include mode.

Copyright © 2012, Juniper Networks, Inc.

It allows the receiver to specify the sources or sources from which it is*not* interested in receiving the multicast group traffic

For each attached network, a multicast router can be either a querier or a nonquerier. A querier router, usually one per subnet, solicits group membership information by transmitting MLD queries. When a host reports to the querier router that it has interested listeners, the querier router forwards the membership information to the rendezvous point (RP) router by means of the receiver's (host's) designated router (DR). This builds the rendezvous-point tree (RPT) connecting the host with interested listeners to the RP router. The RPT is the initial path used by the sender to transmit information to the interested listeners. Non-querier routers do not transmit MLD queries on a subnet but can trasmit them if the querier router goes down.

All MLD-configured routers start up as querier routers on each attached subnet . The non-querier router on the right is the receiver's DR.

To elect the querier router, the routers exchange query messages containing their IPv6 source addresses. If a router hears a query message whose IPv6 source address is numerically lower than its own selected address, it becomes a non-querier. The router on the left has a source address numerically lower than the one on the right and therefore becomes the querier router.

In the practical application of MLD, several routers on a subnet are nonqueriers. If the elected querier router goes down, query messages are exchanged among the remaining routers. The router with the lowest IPv6 source address then becomes the new querier router. Note that the IPv6 Neighbor Discovery Protocol (NDP) implementation drops incoming Neighbor Announcement (NA) messages that have a broadcast or multicast address in the target link-layer address option. This behavior is recommended by RFC 2461.

Configuration option for IPv6 Neighbor Discovery Protocol (NDP) is available. The configuration option is **set protocol neighbor-discovery onlink-subnet-only** command. This option will prevent the device from responding to a Neighbor Solicitation (NS) from a prefix which was not included as one of the device interface prefixes.

Copyright © 2012, Juniper Networks, Inc.

NOTE: The Routing Engine needs to be rebooted after setting this option to remove any possibility of a previous IPv6 entry from remaining in the forwarding-table.

The querier router sends general MLD queries on the **link-scope all-nodes** multicast address **FF02::1** at short intervals to all attached subnets to solicit group membership information. Within the query message is the *maximum response delay* value, specifying the maximum allowed delay for the host to respond with a report message.

Related Documentation

- Multicast Overview on page 353

Multicast PIM and Static RPs

- Understanding PIM and Static RPs on page 367
- Example: Configuring PIM Sparse Mode and RP Static IP Addresses on page 368

Understanding PIM and Static RPs

Protocol Independent Multicast (PIM) sparse mode is the most common multicast protocol used on the Internet. PIM sparse mode is the default mode whenever PIM is configured on any interface of the device. However, because PIM must not be configured on the network management interface, you must disable it on that interface.

Each any-source multicast (ASM) group has a shared tree through which receivers learn about new multicast sources and new receivers learn about all multicast sources. The rendezvous point (RP) router is the root of this shared tree and receives the multicast traffic from the source. To receive multicast traffic from the groups served by the RP, the device must determine the IP address of the RP for the source.

One common way for the device to locate RPs is by static configuration of the IP address of the RP. For information about alternate methods of locating RPs, see the *JunosOS Multicast Protocols Configuration Guide*.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Multicast Overview on page 353
- Example: Configuring PIM Sparse Mode and RP Static IP Addresses on page 368
- PIM Overview in the *Junos OS Multicast Protocols Configuration Guide*
- Understanding PIM Sparse Mode in the *Junos OS Multicast Protocols Configuration Guide*

Example: Configuring PIM Sparse Mode and RP Static IP Addresses

This example shows how to configure PIM sparse mode and RP static IP adresses.

- Requirements on page 368
- Overview on page 368
- Configuration on page 368
- Verification on page 370

Requirements

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its own RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.
7. Configure the SAP and SDP protocols to listen for multicast session announcements. See "Example: Configuring SAP and SDP to Listen for Session Announcements" on page 361.
8. Configure IGMP. See "Example: Configuring IGMP for Multicast" on page 362.

Overview

In this example, you set the interface value to **all** and disable the **ge-0/0/0** interface. Then you configure the IP address of the RP as **192.168.14.27**.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network

Copyright © 2012, Juniper Networks, Inc.

configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set protocols pim interface all
set protocols pim interface ge-0/0/0 disable
set protocols pim rp static address 192.168.14.27
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure PIM sparse mode and the RP static IP address:

1. Configure PIM.

   ```
   [edit]
   user@host# edit protocols pim
   ```

2. Set the interface value.

   ```
   [edit protocols pim]
   user@host# set pim interface all
   ```

3. Disable PIM on the network management interface.

   ```
   [edit protocols pim interface]
   user@host# set pim interface ge-0/0/0 unit 0 disable
   ```

4. Configure RP.

   ```
   [edit]
   user@host# edit protocols pim rp
   ```

5. Configure the IP address of the RP.

   ```
   [edit]
   user@host# set static address 192.168.14.27
   ```

Results

From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show protocols
  pim {
  rp {
  static {
  address 192.168.14.27;
  }
}
interface all;
  interface ge-0/0/0.0 {
  disable;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying SAP and SDP Addresses and Ports on page 370
- Verifying the IGMP Version on page 370
- Verifying the PIM Mode and Interface Configuration on page 370

Verifying SAP and SDP Addresses and Ports

Purpose Verify that SAP and SDP are configured to listen on the correct group addresses and ports.

Action From operational mode, enter the **show sap listen** command.

Verifying the IGMP Version

Purpose Verify that IGMP version 2 is configured on all applicable interfaces.

Action From operational mode, enter the **show igmp interface** command.

Verifying the PIM Mode and Interface Configuration

Purpose Verify that PIM sparse mode is configured on all applicable interfaces.

Action From operational mode, enter the **show pim interfaces** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding PIM and Static RPs on page 367
- PIM Configuration Statements in the *Junos OS Multicast Protocols Configuration Guide*
- Configuring the Static PIM RP Address on the Non-RP Routing Device in the *Junos OS Multicast Protocols Configuration Guide*
- Multicast Configuration Overview on page 359
- Verifying a Multicast Configuration on page 382

PIM Register Messages

- Understanding PIM Register Messages on page 370
- Example: Rejecting Incoming PIM Register Messages on RP Routers on page 371
- Example: Stopping Outgoing PIM Register Messages on a Designated Router on page 374

Understanding PIM Register Messages

When a source in a multicast network becomes active, the source's designated router (DR) encapsulates multicast data packets into a PIM register message and sends them by means of unicast to the rendezvous point (RP) router.

 Copyright © 2012, Juniper Networks, Inc.

To prevent unauthorized groups and sources from registering with an RP router, you can define a routing policy to reject PIM register messages from specific groups and sources and configure the policy on the designated router or the RP router.

- If you configure the reject policy on an RP router, it rejects incoming PIM register messages from the specified groups and sources. The RP router also sends a register stop message by means of unicast to the designated router. On receiving the register stop message, the designated router sends periodic null register messages for the specified groups and sources to the RP router.
- If you configure the reject policy on a designated router, it stops sending PIM register messages for the specified groups and sources to the RP router.

NOTE: If you have configured the reject policy on an RP router, we recommend that you configure the same policy on all the RP routers in your multicast network.

NOTE: If you delete a group and source address from the reject policy configured on an RP router and commit the configuration, the RP router will register the group and source only when the designated router sends a null register message.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Multicast Overview on page 353
- Example: Rejecting Incoming PIM Register Messages on RP Routers on page 371
- Example: Stopping Outgoing PIM Register Messages on a Designated Router on page 374
- Understanding Multicast Message Filters in the *Junos OS Multicast Protocols Configuration Guide*
- Routing Policies Overview on page 387

Example: Rejecting Incoming PIM Register Messages on RP Routers

This example shows how to reject incoming PIM register messages on RP routers.

- Requirements on page 372
- Overview on page 372
- Configuration on page 372
- Verification on page 373

Requirements

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its own RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.
7. Configure the SAP and SDP protocols to listen for multicast session announcements. See "Example: Configuring SAP and SDP to Listen for Session Announcements" on page 361.
8. Configure IGMP. See "Example: Configuring IGMP for Multicast" on page 362.
9. Configure the PIM static RP. See "Understanding PIM and Static RPs" on page 367.

Overview

In this example, you configure the group address as **224.1.1.1/32** and the source address in the group as **10.10.10.1/32**. You set the match action to reject PIM register messages and assign reject-pim-register-msg-rp as the policy on the RP.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options policy-statement reject-pim-register-msg-rp from route-filter
  224.1.1.1/32 exact
set policy-options policy-statement reject-pim-register-msg-rp from source-address-filter
  10.10.10.1/32 exact
set policy-options policy-statement reject-pim-register-msg-rp then reject
set protocols pim rp rp-register-policy reject-pim-register-msg-rp
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To reject the incoming PIM register messages on an RP router:

1. Configure the policy options.

Copyright © 2012, Juniper Networks, Inc.

```
[edit]
user@host# edit policy-options
```

2. Set the group address.

```
[edit policy-options]
user@host# set policy statement reject-pim-register-msg-rp from route-filter
  224.1.1.1/32 exact
```

3. Set the source address.

```
[edit policy-options]
user@host# set policy statement reject-pim-register-msg-rp from
  source-address-filter 10.10.10.1/32 exact
```

4. Set the match action.

```
[edit policy-options]
user@host# set policy statement reject-pim-register-msg-rp then reject
```

5. Configure the protocol.

```
[edit]
user@host# edit protocols pim rp
```

6. Assign the policy.

```
[edit]
user@host# set rp-register-policy reject-pim-register-msg-rp
```

Results From configuration mode, confirm your configuration by entering the **show policy-options** and **show protocols pim** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show policy-options
  policy-statement reject-pim-register-msg-rp {
  from {
  route-filter 224.1.1.1/32 exact;
  source-address-filter 10.10.10.1/32 exact;
  }
  then reject;
}
[edit]
  user@host# show protocols pim
  rp {
  rp-register-policy reject-pim-register-msg-rp;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying SAP and SDP Addresses and Ports on page 374
- Verifying the IGMP Version on page 374

- Verifying the PIM Mode and Interface Configuration on page 374
- Verifying the PIM Register Messages on page 374

Verifying SAP and SDP Addresses and Ports

Purpose Verify that SAP and SDP are configured to listen on the correct group addresses and ports.

Action From operational mode, enter the **show sap listen** command.

Verifying the IGMP Version

Purpose Verify that IGMP version 2 is configured on all applicable interfaces.

Action From operational mode, enter the **show igmp interface** command.

Verifying the PIM Mode and Interface Configuration

Purpose Verify that PIM sparse mode is configured on all applicable interfaces.

Action From operational mode, enter the **show pim interfaces** command.

Verifying the PIM Register Messages

Purpose Verify whether the rejected policy on the RP router is enabled.

Action From operational mode, enter the **show policy-options** and **show protocols pim** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding PIM Register Messages on page 370
- Example: Stopping Outgoing PIM Register Messages on a Designated Router on page 374
- Configuring Register Message Filters on a PIM RP and DR in the *Junos OS Multicast Protocols Configuration Guide*
- Multicast Configuration Overview on page 359
- Verifying a Multicast Configuration on page 382

Example: Stopping Outgoing PIM Register Messages on a Designated Router

This example shows how to stop outgoing PIM register messages on a designated router.

- Requirements on page 375
- Overview on page 375
- Configuration on page 375
- Verification on page 377

 Copyright © 2012, Juniper Networks, Inc.

Requirements

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its own RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.
7. Configure the SAP and SDP protocols to listen for multicast session announcements. See "Example: Configuring SAP and SDP to Listen for Session Announcements" on page 361.
8. Configure IGMP. See "Example: Configuring IGMP for Multicast" on page 362.
9. Configure the PIM static RP. See "Understanding PIM and Static RPs" on page 367.
10. Filter PIM register messages from unauthorized groups and sources. See "Example: Rejecting Incoming PIM Register Messages on RP Routers" on page 371.

Overview

In this example, you configure the group address as **224.2.2.2/32** and the source address in the group as **20.20.20.1/32**. You set the match action to not send PIM register messages for the group and source address. Then you configure the policy on the designated router to **stop-pim-register-msg-dr**.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options policy-statement stop-pim-register-msg-dr from route-filter
  224.2.2.2/32 exact
set policy-options policy-statement stop-pim-register-msg-dr from source-address-filter
  20.20.20.1/32 exact
set policy-options policy-statement stop-pim-register-msg-dr then reject
set protocols pim rp dr-register-policy stop-pim-register-msg-dr
```

Copyright © 2012, Juniper Networks, Inc.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To stop outgoing PIM register messages on a designated router:

1. Configure the policy options.

```
[edit]
user@host# edit policy-options
```

2. Set the group address.

```
[edit policy-options]
user@host# set policy statement stop-pim-register-msg-dr from route-filter
  224.2.2.2/32 exact
```

3. Set the source address.

```
[edit policy-options]
user@host# set policy statement stop-pim-register-msg-dr from
  source-address-filter 20.20.20.1/32 exact
```

4. Set the match action.

```
[edit policy-options]
user@host# set policy statement stop-pim-register-msg-dr then reject
```

5. Assign the policy.

```
[edit]
user@host# set dr-register-policy stop-pim-register-msg-dr
```

Results

From configuration mode, confirm your configuration by entering the **show policy-options** and **show protocols** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show policy-options
  policy-statement stop-pim-register-msg-dr {
  from {
  route-filter 224.2.2.2/32 exact;
  source-address-filter 20.20.20.1/32 exact;
  }
  then reject;
}
[edit]
  user@host# show protocols
  pim {
  rp {
  dr-register-policy stop-pim-register-msg-dr;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Copyright © 2012, Juniper Networks, Inc.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying SAP and SDP Addresses and Ports on page 377
- Verifying the IGMP Version on page 377
- Verifying the PIM Mode and Interface Configuration on page 377
- Verifying the PIM RP Configuration on page 377

Verifying SAP and SDP Addresses and Ports

Purpose Verify that SAP and SDP are configured to listen on the correct group addresses and ports.

Action From operational mode, enter the **show sap listen** command.

Verifying the IGMP Version

Purpose Verify that IGMP version 2 is configured on all applicable interfaces.

Action From operational mode, enter the **show igmp interface** command.

Verifying the PIM Mode and Interface Configuration

Purpose Verify that PIM sparse mode is configured on all applicable interfaces.

Action From operational mode, enter the **show pim interfaces** command.

Verifying the PIM RP Configuration

Purpose Verify that the PIM RP is statically configured with the correct IP address.

Action From operational mode, enter the **show pim rps** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding PIM Register Messages on page 370
- Configuring Register Message Filters on a PIM RP and DR in the *Junos OS Multicast Protocols Configuration Guide*
- Multicast Configuration Overview on page 359

PIM RPF Routing Tables

- Understanding PIM RPF Routing Tables on page 378
- Example: Configuring a PIM RPF Routing Table on page 378

Copyright © 2012, Juniper Networks, Inc.

Understanding PIM RPF Routing Tables

By default, PIM uses **inet.0** as its reverse-path forwarding (RPF) routing table group. PIM uses an RPF routing table group to resolve its RPF neighbor for a particular multicast source address and for the RP address. PIM can optionally use **inet.2** as its RPF routing table group. The **inet.2** routing table is organized more efficiently for RPF checks.

Once configured, the RPF routing table must be applied to a PIM as a routing table group.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Multicast Overview on page 353
- Example: Configuring a PIM RPF Routing Table on page 378
- Example: Configuring a Dedicated PIM RPF Routing Table in the *Junos OS Multicast Protocols Configuration Guide*

Example: Configuring a PIM RPF Routing Table

This example shows how to configure and apply a PIM RPF routing table.

- Requirements on page 378
- Overview on page 379
- Configuration on page 379
- Verification on page 380

Requirements

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.
7. Configure the SAP and SDP protocols to listen for multicast session announcements. See "Example: Configuring SAP and SDP to Listen for Session Announcements" on page 361.
8. Configure IGMP. See "Example: Configuring IGMP for Multicast" on page 362.

Copyright © 2012, Juniper Networks, Inc.

9. Configure the PIM static RP. See "Understanding PIM and Static RPs" on page 367.
10. Filter PIM register messages from unauthorized groups and sources. See "Example: Rejecting Incoming PIM Register Messages on RP Routers" on page 371 and "Example: Stopping Outgoing PIM Register Messages on a Designated Router" on page 374.

Overview

In this example, you name the new RPF routing table group **multicast-rfp-rib** and use **inet.2** for its export as well as its import routing table. Then you create a routing table group for the interface routes and name the RPF **if-rib**. Finally, you use **inet.2** and **inet.0** for its import routing tables, and add the new interface routing table group to the interface routes.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set routing-options rib-groups multicast-rpf-rib export-rib inet.2
set routing-options rib-groups multicast-rpf-rib import-rib inet.2
set protocols pim rib-group multicast-rpf-rib
set routing-options rib-groups if-rib import-rib inet.2
set routing-options rib-groups if-rib import-rib inet.0
set routing-options interface-routes rib-group inet if-rib
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure the PIM RPF routing table:

1. Configure a routing option and a group.

   ```
   [edit]
   user@host# edit routing-options rib-groups
   ```

2. Configure a name.

   ```
   [edit routing-options rib-groups]
   user@host# set multicast-rpf-rib export-rib inet.2
   ```

3. Create a new group for the RPF routing table.

   ```
   [edit routing-options rib-groups]
   user@host# set multicast-rpf-rib import-rib inet.2
   ```

4. Apply the new RPF routing table.

   ```
   [edit protocols pim]
   user@host# set rib-group multicast-rpf-rib
   ```

5. Create a routing table group for the interface routes.

   ```
   [edit]
   user@host# edit routing-options rib-groups
   ```

Copyright © 2012, Juniper Networks, Inc.

6. Configure a name for import routing table.

```
[edit routing-options rib-groups]
user@host# set if-rib import-rib inet.2
user@host# set if-rib import-rib inet.0
```

7. Set group to interface routes.

```
[edit routing-options interface-routes]
user@host# set rib-group inet if-rib
```

Results From configuration mode, confirm your configuration by entering the **show protocols** and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show protocols
  pim {
  rib-group inet multicast-rpf-rib;
}
[edit]
  user@host# show routing-options
  interface-routes {
  rib-group inet if-rib;
  }
  static {
  route 0.0.0.0/0 next-hop 10.100.37.1;
  }
  rib-groups {
  multicast-rpf-rib {
  export-rib inet.2;
  import-rib inet.2;
  }
  if-rib {
  import-rib [ inet.2 inet.0 ];
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying SAP and SDP Addresses and Ports on page 380
- Verifying the IGMP Version on page 381
- Verifying the PIM Mode and Interface Configuration on page 381
- Verifying the PIM RP Configuration on page 381
- Verifying the RPF Routing Table Configuration on page 381

Verifying SAP and SDP Addresses and Ports

Purpose Verify that SAP and SDP are configured to listen on the correct group addresses and ports.

 Copyright © 2012, Juniper Networks, Inc.

Action From operational mode, enter the **show sap listen** command.

Verifying the IGMP Version

Purpose Verify that IGMP version 2 is configured on all applicable interfaces.

Action From operational mode, enter the **show igmp interface** command.

```
user@host> show igmp interface
Interface: ge-0/0/0.0
    Querier: 192.168.4.36
    State:         Up Timeout:     197 Version:  2 Groups:      0

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Derived Parameters:
IGMP Membership Timeout: 260.0
IGMP Other Querier Present Timeout: 255.0
```

Verifying the PIM Mode and Interface Configuration

Purpose Verify that PIM sparse mode is configured on all applicable interfaces.

Action From operational mode, enter the **show pim interfaces** command.

Verifying the PIM RP Configuration

Purpose Verify that the PIM RP is statically configured with the correct IP address.

Action From operational mode, enter the **show pim rps** command.

Verifying the RPF Routing Table Configuration

Purpose Verify that the PIM RPF routing table is configured correctly.

Action From operational mode, enter the **show multicast rpf** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding PIM Register Messages on page 370
- Example: Configuring a Dedicated PIM RPF Routing Table in the *Junos OS Multicast Protocols Configuration Guide*
- Multicast Configuration Overview on page 359
- Verifying a Multicast Configuration on page 382

Copyright © 2012, Juniper Networks, Inc.

Verifying a Multicast Configuration

To verify a multicast configuration, perform these tasks:

- Verifying SAP and SDP Addresses and Ports on page 382
- Verifying the IGMP Version on page 382
- Verifying the PIM Mode and Interface Configuration on page 383
- Verifying the PIM RP Configuration on page 383
- Verifying the RPF Routing Table Configuration on page 384

Verifying SAP and SDP Addresses and Ports

Purpose Verify that SAP and SDP are configured to listen on the correct group addresses and ports.

Action From the CLI, enter the **show sap listen** command.

Sample Output

```
user@host> show sap listen
Group Address   Port
224.2.127.254   9875
```

Meaning The output shows a list of the group addresses and ports that SAP and SDP listen on. Verify the following information:

- Each group address configured, especially the default **224.2.127.254**, is listed.
- Each port configured, especially the default **9875**, is listed.

Verifying the IGMP Version

Purpose Verify that IGMP version 2 is configured on all applicable interfaces.

Action From the CLI, enter the **show igmp interface** command.

Sample Output

```
user@host> show igmp interface
Interface: ge-0/0/0.0
    Querier: 192.168.4.36
    State:         Up Timeout:     197 Version:  2 Groups:      0

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Derived Parameters:
```

 Copyright © 2012, Juniper Networks, Inc.

```
IGMP Membership Timeout: 260.0
IGMP Other Querier Present Timeout: 255.0
```

Meaning The output shows a list of the interfaces that are configured for IGMP. Verify the following information:

- Each interface on which IGMP is enabled is listed.
- Next to **Version**, the number 2 appears.

Verifying the PIM Mode and Interface Configuration

Purpose Verify that PIM sparse mode is configured on all applicable interfaces.

Action From the CLI, enter the **show pim interfaces** command.

Sample Output

```
user@host> show pim interfaces
Instance: PIM.master
Name                    Stat Mode       IP V State Count DR address
lo0.0                   Up   Sparse      4 2 DR        0 127.0.0.1
pime.32769              Up   Sparse      4 2 P2P       0
```

Meaning The output shows a list of the interfaces that are configured for PIM. Verify the following information:

- Each interface on which PIM is enabled is listed.
- The network management interface, either **ge–0/0/0** or **fe–0/0/0**, is *not* listed.
- Under **Mode**, the word **Sparse** appears.

Verifying the PIM RP Configuration

Purpose Verify that the PIM RP is statically configured with the correct IP address.

Action From the CLI, enter the **show pim rps** command.

Sample Output

```
user@host> show pim rps
Instance: PIM.master
Address family INET
RP address      Type      Holdtime Timeout Active groups Group prefixes
192.168.14.27   static           0    None             2 224.0.0.0/4
```

Meaning The output shows a list of the RP addresses that are configured for PIM. At least one RP must be configured. Verify the following information:

- The configured RP is listed with the proper IP address.
- Under **Type**, the word **static** appears.

Verifying the RPF Routing Table Configuration

Purpose Verify that the PIM RPF routing table is configured correctly.

Action From the CLI, enter the **show multicast rpf** command.

Sample Output

```
user@host> show multicast rpf
Multicast RPF table: inet.0 , 2 entries...
```

Meaning The output shows the multicast RPF table that is configured for PIM. If no multicast RPF routing table is configured, RPF checks use **inet.0**. Verify the following information:

- The configured multicast RPF routing table is **inet.0**.
- The **inet.0** table contains entries.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Multicast Configuration Overview on page 359
- show sap listen in the *Junos OS Routing Protocols and Policies Command Reference*
- show igmp interface in the *Junos OS Routing Protocols and Policies Command Reference*
- show pim interfaces in the *Junos OS Routing Protocols and Policies Command Reference*
- show pim rps in the *Junos OS Routing Protocols and Policies Command Reference*
- show multicast rpf in the *Junos OS Routing Protocols and Policies Command Reference*

 Copyright © 2012, Juniper Networks, Inc.

PART 2

Routing Policies and Stateless Firewall Filters

- Routing Policies on page 387
- Stateless Firewall Filters on page 419

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 14

Routing Policies

- Routing Policies Overview on page 387
- Routing Policies Configuration Overview on page 388
- Routing Policies on page 389
- Routing Policy Terms on page 390
- Routing Policy Match Conditions and Actions on page 392
- Routing Policy Damping Parameters on page 408
- ECMP Flow-Based Forwarding on page 412

Routing Policies Overview

A routing policy has a major impact on the flow of routing information or packets within and through the device. The match conditions and actions allow you to configure a customized policy to fit your needs.

Routing protocols send information about routes to a router's neighbors. This information is processed and used to create routing tables, which are then distilled into forwarding tables. Routing policies control the flow of information between the routing protocols and the routing tables and between the routing tables and the forwarding tables. Using policies, you can determine which routes are advertised, specify which routes are imported into the routing table, and modify routes to control which routes are added to the forwarding table.

To create a routing policy, you configure criteria against which routes are compared, and the action that is performed if the criteria are met.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Configuration Overview on page 388
- Routing Overview on page 3

Routing Policies Configuration Overview

To configure a routing policy:

1. Determine what you want to accomplish with the policy, and thoroughly understand how to achieve your goal using the various match conditions and actions.
2. Make certain that you understand the default policies and actions for the policy you are configuring.
3. Configure an interface on the router. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
4. Configure an Interior Gateway Protocol (IGP) and Border Gateway Protocol (BGP), if necessary. See:
   - RIP Configuration Overview on page 137
   - OSPF Configuration Overview on page 165
   - Example: Configuring IS-IS on page 219
   - BGP Configuration Overview on page 232
5. Configure the router interface to reject or accept routes, if necessary.
6. Configure static routes, if necessary. See "Example: Configuring a Basic Set of Static Routes" on page 24.
7. Name the policy. See "Example: Creating a Routing Policy" on page 389.
8. Configure the policy term. See "Example: Creating a Routing Policy Term" on page 391.
9. (Optional) Reject useless routes. See "Example: Rejecting Known Invalid Routes" on page 397.
10. (Optional) Advertise additional routes. See "Example: Injecting OSPF Routes into the BGP Routing Table" on page 402.
11. (Optional) Create a forwarding class. See "Example: Grouping Source and Destination Prefixes into a Forwarding Class" on page 399.
12. (Optional) Make a route less preferable to BGP. See "Example: Configuring a Routing Policy to Prepend the AS Path" on page 406.
13. (Optional) Suppress route information. See "Example: Configuring Damping Parameters" on page 409.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Overview on page 387
- Minimum Routing Policy Configuration in the *Junos OS Policy Framework Configuration Guide*
- Testing Routing Policies in the *Junos OS Policy Framework Configuration Guide*

Copyright © 2012, Juniper Networks, Inc.

Routing Policies

- Understanding Routing Policies on page 389
- Example: Creating a Routing Policy on page 389

Understanding Routing Policies

Each routing policy is identified by a policy name. The name can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. To include spaces in the name, enclose the entire name in double quotation marks. Each routing policy name must be unique within a configuration.

Once a policy is created and named, it must be applied before it is active. You apply routing policies using the **import** and **export** statements at the **protocols>*protocol-name*** level in the configuration hierarchy.

In the **import** statement, you list the name of the routing policy to be evaluated when routes are imported into the routing table from the routing protocol.

In the **export** statement, you list the name of the routing policy to be evaluated when routes are being exported from the routing table into a dynamic routing protocol. Only active routes are exported from the routing table.

To specify more than one policy and create a policy chain, you list the policies using a space as a separator. If multiple policies are specified, the policies are evaluated in the order in which they are specified. As soon as an accept or reject action is executed, the policy chain evaluation ends.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Overview on page 387
- Example: Creating a Routing Policy on page 389
- Router Flows Affected by Policies in the *Junos OS Policy Framework Configuration Guide*

Example: Creating a Routing Policy

This example shows how to create a simple routing policy.

- Requirements on page 389
- Overview on page 390
- Configuration on page 390
- Verification on page 390

Requirements

Before you begin, determine what you want to accomplish with the policy, configure router interfaces, and configure routing protocols, as explained in "Routing Policies Configuration Overview" on page 388.

Overview

In this example, you create a routing policy called policy1.

Configuration

Step-by-Step Procedure

To create a routing policy:

1. Create the routing policy.

```
[edit]
user@host# set policy-options policy-statement policy1
```

2. Commit the configuration if you are done configuring the device.

```
[edit]
user@host# commit
```

NOTE: The policy does not take effect until you apply it.

Verification

To verify your configuration, use the **show policy-options** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Configuration Overview on page 388
- Understanding Routing Policies on page 389

Routing Policy Terms

- Understanding Routing Policy Terms on page 390
- Example: Creating a Routing Policy Term on page 391

Understanding Routing Policy Terms

Routing policies are made up of one or more terms. Each routing policy term is identified by a term name. The name can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. To include spaces in the name, enclose the entire name in double quotation marks.

Each term contains a set of match conditions and a set of actions:

- Match conditions are criteria that a route must match before the actions can be applied. If a route matches all criteria, one or more actions are applied to the route.
- Actions specify whether to accept or reject the route, control how a series of policies are evaluated, and manipulate the characteristics associated with a route.

Copyright © 2012, Juniper Networks, Inc.

Generally, a router compares a route against the match conditions of each term in a routing policy, starting with the first and moving through the terms in the order in which they are defined, until a match is made and an explicitly configured or default action of **accept** or **reject** is taken. If none of the terms in the policy match the route, the router compares the route against the next policy, and so on, until either an action is taken or the default policy is evaluated.

If none of the match conditions of each term evaluates to true, the final action is executed. The final action is defined in an unnamed term. Additionally, you can define a default action (either **accept** or **reject**) that overrides any action intrinsic to the protocol.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Overview on page 387
- Example: Creating a Routing Policy Term on page 391

Example: Creating a Routing Policy Term

This example shows how to create a routing policy term.

- Requirements on page 391
- Overview on page 391
- Configuration on page 391
- Verification on page 392

Requirements

Before you begin, determine what you want to accomplish with the policy, configure router interfaces, and configure routing protocols, as explained in "Routing Policies Configuration Overview" on page 388.

Overview

In this example, you create a routing policy called policy1 and a term for the policy called term1.

Configuration

Step-by-Step Procedure

To configure a routing policy term:

1. Create the routing policy.

```
[edit]
user@host# edit policy-options policy-statement policy1
```

2. Create the policy term.

```
[edit policy-options policy-statement policy1]
user@host# set term term1
```

3. Commit the configuration if you are done configuring the device.

```
[edit policy-options policy-statement policy1]
user@host# commit
```

NOTE: The policy does not take effect until you apply it.

Verification

To verify your configuration, use the **show policy-options** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Configuration Overview on page 388
- Understanding Routing Policy Terms on page 390

Routing Policy Match Conditions and Actions

- Understanding Routing Policy Match Conditions and Actions on page 392
- Route-Based Match Conditions on page 396
- Protocol-Based Match Conditions on page 401
- Autonomous System Path-Based Actions on page 405

Understanding Routing Policy Match Conditions and Actions

A match condition defines the criteria that a route must match for an action to take place. Each term can have one or more match conditions. If a route matches all the match conditions for a particular term, the actions defined for that term are processed.

This topic contains the following sections:

- Match Conditions on page 392
- Actions on page 394

Match Conditions

Each term can consist of two statements, **from** and **to**, that define match conditions:

- In the **from** statement, you define the criteria that an *incoming* route must match. You can specify one or more match conditions. If you specify more than one, all conditions must match the route for a match to occur.
- In the **to** statement, you define the criteria that an *outgoing* route must match. You can specify one or more match conditions. If you specify more than one, all conditions must match the route for a match to occur.

The order of match conditions in a term is not important, because a route must match all match conditions in a term for an action to be taken.

Table 10 on page 393 summarizes key routing policy match conditions.

Copyright © 2012, Juniper Networks, Inc.

Table 10: Summary of Key Routing Policy Match Conditions

Match Condition	Description
aggregate-contributor	Matches routes that are contributing to a configured aggregate. This match condition can be used to suppress a contributor in an aggregate route.
area *area-id*	Matches a route learned from the specified OSPF area during the exporting of OSPF routes into other protocols.
as-path *name*	Matches the name of the path regular expression of an autonomous systems (AS). BGP routes whose AS path matches the regular expression are processed.
color *preference*	Matches a color value. You can specify preference values that are finer-grained than those specified in the ***preference*** match conditions. The **color** value can be a number from **0** through **4,294,967,295** ($2^{32} - 1$). A lower number indicates a more preferred route.
community	Matches the name of one or more communities. If you list more than one name, only one name needs to match for a match to occur. (The matching is effectively a logical OR operation.)
external [type *metric-type***]**	Matches external OSPF routes, including routes exported from one level to another. In this match condition, **type** is an optional keyword. The **metric-type** value can be either **1** or **2**. When you do not specify **type**, this condition matches all external routes.
interface *interface-name*	Matches the name or IP address of one or more router interfaces. Use this condition with protocols that are interface-specific. For example, do not use this condition with internal BGP (IBGP). Depending on where the policy is applied, this match condition matches routes learned from or advertised through the specified interface.
internal	Matches a routing policy against the internal flag for simplified next-hop self policies.
level *level*	Matches the IS-IS level. Routes that are from the specified level or are being advertised to the specified level are processed.
local-preference *value*	Matches a BGP local preference attribute. The preference value can be from **0** through **4,294,967,295** ($2^{32} - 1$).
metric *metric* **metric2** *metric*	Matches a metric value. The **metric** value corresponds to the multiple exit discriminator (MED), and **metric2** corresponds to the IGP metric if the BGP next hop runs back through another route.
neighbor *address*	Matches the address of one or more neighbors (peers). For BGP export policies, the address can be for a directly connected or indirectly connected peer. For all other protocols, the address is for the neighbor from which the advertisement is received.
next-hop *address*	Matches the next-hop address or addresses specified in the routing information for a particular route. For BGP routes, matches are performed against each protocol next hop.

Copyright © 2012, Juniper Networks, Inc.

Table 10: Summary of Key Routing Policy Match Conditions *(continued)*

Match Condition	Description
origin *value*	Matches the BGP origin attribute, which is the origin of the AS path information. The value can be one of the following: • **egp**—Path information originated from another AS. • **igp**—Path information originated from within the local AS. • **incomplete**—Path information was learned by some other means.
preference *preference* **preference2** *preference*	Matches the preference value. You can specify a primary preference value (**preference**) and a secondary preference value (**preference2**). The preference value can be a number from **0** through **4,294,967,295** ($2^{32}-1$). A lower number indicates a more preferred route.
protocol *protocol*	Matches the name of the protocol from which the route was learned or to which the route is being advertised. It can be one of the following: **aggregate**, **bgp**, **direct**, **dvmrp**, **isis**, **local**, **ospf**, **pim-dense**, **pim-sparse**, **rip**, **ripng**, or **static**.
route-type *value*	Matches the type of route. The value can be either **external** or **internal**.

Actions

An action defines what the router does with the route when the route matches all the match conditions in the **from** and **to** statements for a particular term. If a term does not have **from** and **to** statements, all routes are considered to match and the actions apply to all routes.

Each term can have one or more of the following types of actions. The actions are configured under the **then** statement.

- Flow control actions, which affect whether to accept or reject the route and whether to evaluate the next term or routing policy
- Actions that manipulate route characteristics
- Trace action, which logs route matches

If you do not specify an action, one of the following results occurs:

- The next term in the routing policy, if one exists, is evaluated.
- If the routing policy has no more terms, the next routing policy, if one exists, is evaluated.
- If there are no more terms or routing policies, the accept or reject action specified by the default policy is executed.

Table 11 on page 394 summarizes the routing policy actions.

Table 11: Summary of Key Routing Policy Actions

Action	Description
Flow Control Actions	These actions control the flow of routing information into and out of the routing table.

Copyright © 2012, Juniper Networks, Inc.

Table 11: Summary of Key Routing Policy Actions *(continued)*

Action	Description
accept	Accepts the route and propagates it. After a route is accepted, no other terms in the routing policy and no other routing policies are evaluated.
reject	Rejects the route and does not propagate it. After a route is rejected, no other terms in the routing policy and no other routing policies are evaluated.
next term	Skips to and evaluates the next term in the same routing policy. Any **accept** or **reject** action specified in the **then** statement is ignored. Any actions specified in the **then** statement that manipulate route characteristics are applied to the route.
next policy	Skips to and evaluates the next routing policy. Any **accept** or **reject** action specified in the **then** statement is ignored. Any actions specified in the **then** statement that manipulate route characteristics are applied to the route.
Route Manipulation Actions	These actions manipulate the route characteristics.
as-path-prepend *as-path*	Appends one or more AS numbers at the beginning of the AS path. If you are specifying more than one AS number, include the numbers in quotation marks. The AS numbers are added after the local AS number has been added to the path. This action adds AS numbers to AS sequences only, not to AS sets. If the existing AS path begins with a confederation sequence or set, the appended AS numbers are placed within a confederation sequence. Otherwise, the appended AS numbers are placed with a nonconfederation sequence.
as-path-expand last-as count *n*	Extracts the last AS number in the existing AS path and appends that AS number to the beginning of the AS path *n* times. Replace *n* with a number from **1** through **32**. The AS numbers are added after the local AS number has been added to the path. This action adds AS numbers to AS sequences only, not to AS sets. If the existing AS path begins with a confederation sequence or set, the appended AS numbers are placed within a confederation sequence. Otherwise, the appended AS numbers are placed with a nonconfederation sequence.
class *class-name*	Applies the specified class-of-service (CoS) parameters to routes installed into the routing table.
color *preference* **color2** *preference*	Sets the preference value to the specified value. The **color** and **color2** preference values can be a number from **0** through **4,294,967,295** ($2^{32} - 1$). A lower number indicates a more preferred route.
damping *name*	Applies the specified route-damping parameters to the route. These parameters override BGP's default damping parameters. This action is useful only in import policies.
local-preference *value*	Sets the BGP local preference attribute. The preference can be a number from **0** through **4,294,967,295** ($2^{32} - 1$).

Table 11: Summary of Key Routing Policy Actions *(continued)*

Action	Description
metric *metric* metric2 *metric* metric3 *metric* metric4 *metric*	Sets the metric. You can specify up to four metric values, starting with **metric** (for the first metric value) and continuing with **metric2**, **metric3**, and **metric4**. For BGP routes, **metric** corresponds to the MED, and **metric2** corresponds to the IGP metric if the BGP next hop loops through another router.
next-hop *address*	Sets the next hop. If you specify ***address*** as **self**, the next-hop address is replaced by one of the local router's addresses. The advertising protocol determines which address to use.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Overview on page 387
- Understanding Route-Based Match Conditions on page 396
- Understanding Protocol-Based Match Conditions on page 402
- Understanding Autonomous System Path-Based Actions on page 405
- Configuring Match Conditions in Routing Policy Terms in the *Junos OS Policy Framework Configuration Guide*
- Configuring Actions in Routing Policy Terms in the *Junos OS Policy Framework Configuration Guide*

Route-Based Match Conditions

- Understanding Route-Based Match Conditions on page 396
- Example: Rejecting Known Invalid Routes on page 397
- Example: Grouping Source and Destination Prefixes into a Forwarding Class on page 399

Understanding Route-Based Match Conditions

You can specify known invalid (“bad”) routes to ignore by specifying matches on destination prefixes. When specifying a destination prefix, you can specify an exact match with a specific route, or a less precise match by using match types. You can configure either a common reject action that applies to the entire list, or an action associated with each prefix.

Additionally, you can specify that “good” routes be processed in a particular way. For instance, you can group traffic from specific source or destination addresses into forwarding classes to be processed using the class of service (CoS) feature.

Table 12 on page 397 lists route list match types.

Copyright © 2012, Juniper Networks, Inc.

Table 12: Route List Match Types

Match Type	Match Conditions
exact	The route shares the same most-significant bits (described by ***prefix-length***), and ***prefix-length*** is equal to the route's prefix length.
longer	The route shares the same most-significant bits (described by ***prefix-length***), and ***prefix-length*** is greater than the route's prefix length.
orlonger	The route shares the same most-significant bits (described by ***prefix-length***), and ***prefix-length*** is equal to or greater than the route's prefix length.
prefix-length-range ***prefix-length2-prefix-length3***	The route shares the same most-significant bits (described by ***prefix-length***), and the route's prefix length falls between ***prefix-length2*** and ***prefix-length3***, inclusive.
through ***destination-prefix***	All the following are true: • The route shares the same most-significant bits (described by ***prefix-length***) of the first destination prefix. • The route shares the same most-significant bits (described by ***prefix-length***) of the second destination prefix for the number of bits in the prefix length. • The number of bits in the route's prefix length is less than or equal to the number of bits in the second prefix. You do not use the **through** match type in most routing policy configurations. See the *Junos Policy Framework Configuration Guide*.
upto ***prefix-length2***	The route shares the same most-significant bits (described by ***prefix-length***) and the route's prefix length falls between ***prefix-length*** and ***prefix-length2***.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Routing Policy Match Conditions and Actions on page 392
- Example: Rejecting Known Invalid Routes on page 397
- Example: Grouping Source and Destination Prefixes into a Forwarding Class on page 399
- Routing Policies Configuration Overview on page 388
- *Junos OS Class of Service Configuration Guide for Security Devices*

Example: Rejecting Known Invalid Routes

This example shows how to create route-based match conditions for a routing policy.

- Requirements on page 398
- Overview on page 398
- Configuration on page 398
- Verification on page 399

Requirements

Before you begin, configure router interfaces and configure routing protocols, as explained in "Routing Policies Configuration Overview" on page 388.

Overview

In this example, you create a policy called rejectpolicy1 that rejects routes with a mask of /8 and greater (/8, /9, /10, and so on) that have the first 8 bits set to 0. This policy also accepts routes less than 8 bits in length by creating a mask of 0/0 up to /7.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options policy-statement rejectpolicy1 term rejectterm1 from route-filter
   0.0.0.0/0 upto /7 accept
set policy-options policy-statement rejectpolicy1 term rejectterm1 from route-filter
   0.0.0.0/8 orlonger reject
set policy-options policy-statement test term 1 from protocol direct
```

Step-by-Step Procedure

To create a policy that rejects known invalid routes:

1. Create the routing policy.

```
[edit]
user@host# edit policy-options policy-statement rejectpolicy1
```

2. Create the policy term.

```
[edit policy-options policy-statement rejectpolicy1]
user@host# edit term rejectterm1
```

3. Create a mask that specifies which routes to accept.

```
[edit policy-options policy-statement rejectpolicy1 term rejectterm1]
user@host# set from route-filter 0/0 upto /7 accept
```

4. Create a mask that specifies which routes to reject.

```
[edit policy-options policy-statement rejectpolicy1 term rejectterm1]
user@host# set from route-filter 0/8 orlonger reject
```

Results

Confirm your configuration by entering the **show policy-options** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show policy-options
policy-statement rejectpolicy1 {
   term rejectterm1 {
      from {
         route-filter 0.0.0.0/0 upto /7 accept;
         route-filter 0.0.0.0/8 orlonger reject;
      }
```


Copyright © 2012, Juniper Networks, Inc.

```
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the Route-Based Match Conditions on page 399

Verifying the Route-Based Match Conditions

Purpose Verify that the policy and term are configured on the device with the appropriate route-based match conditions.

Action From operational mode, enter the **show policy-options** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Configuration Overview on page 388
- Understanding Route-Based Match Conditions on page 396
- Example: Grouping Source and Destination Prefixes into a Forwarding Class on page 399

Example: Grouping Source and Destination Prefixes into a Forwarding Class

This example shows how to group source and destination prefixes into a forwarding class.

- Requirements on page 399
- Overview on page 399
- Configuration on page 400
- Verification on page 401

Requirements

Before you begin, configure router interfaces and configure routing protocols, as explained in "Routing Policies Configuration Overview" on page 388.

Overview

In this example, you configure a routing policy called policy1and a corresponding routing term called term1. Within the term, you configure the route filter to include source routes greater than or equal to 10.210.0.0/16 and destination routes greater than or equal to 10.215.0.0/16. Then you group the source and destination prefixes into a forwarding class called forwarding-class1 and apply policy1 to the forwarding table. The routing policy is evaluated when routes are being exported from the routing table into the forwarding table. Only the active routes are exported from the routing table.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options policy-statement policy1 term term1 from route-filter 10.210.0.0/16
  orlonger
set policy-options policy-statement policy1 term term1 from route-filter 10.215.0.0/16
  orlonger
set policy-options policy-statement policy1 term term1 then forwarding-class
  forwarding-class1
set routing-options forwarding-table export policy1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To group source and destination prefixes in a forwarding class:

1. Create the routing policy.

   ```
   [edit]
   user@host# edit policy-options policy-statement policy1
   ```

2. Create the routing term.

   ```
   [edit policy-options policy-statement policy1]
   user@host# edit term term1
   ```

3. Specify the source routes to include in the route filter.

   ```
   [edit policy-options policy-statement policy1 term term1]
   user@host# set from route-filter 10.210.0.0/16 orlonger
   ```

4. Specify the destination routes to include in the route filter.

   ```
   [edit policy-options policy-statement policy1 term term1]
   user@host# set from route-filter 10.215.0.0/16 orlonger
   ```

5. Group the source and destination prefixes into the forwarding class.

   ```
   [edit policy-options policy-statement policy1 term term1]
   user@host# set then forwarding-class forwarding-class1
   ```

6. Apply the routing policy to the forwarding table.

   ```
   [edit]
   user@host# set routing-options forwarding-table export policy1
   ```

NOTE: You can refer to the same routing policy one or more times in the same or different export statement.

Copyright © 2012, Juniper Networks, Inc.

Results Confirm your configuration by entering the **show policy-options** and **show routing-options** commands from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show policy-options
policy-statement policy1 {
   term term1 {
      from {
         route-filter 10.210.0.0/16 orlonger;
         route-filter 10.215.0.0/16 orlonger;
      }
      then forwarding-class forwarding-class1;
   }
}
```

```
user@host# show routing-options
forwarding-table {
   export policy1;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the Forwarding Class on page 401
- Verifying the Routing Policy on page 401

Verifying the Forwarding Class

Purpose Verify that the forwarding table is applied to the routing policy.

Action From operational mode, enter the **show routing-options** command.

Verifying the Routing Policy

Purpose Verify that the policy and term are configured on the device with the appropriate routes included in the forwarding class.

Action From operational mode, enter the **show policy-options** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Configuration Overview on page 388
- Understanding Route-Based Match Conditions on page 396
- Example: Rejecting Known Invalid Routes on page 397

Protocol-Based Match Conditions

- Understanding Protocol-Based Match Conditions on page 402
- Example: Injecting OSPF Routes into the BGP Routing Table on page 402

Copyright © 2012, Juniper Networks, Inc.

Understanding Protocol-Based Match Conditions

You can specify a match condition for policies based on protocols by naming a protocol from which the route is learned or to which the route is being advertised. You can specify one of the following protocols: aggregate, BGP, direct, DVMRP, IS-IS, local, OSPF, PIM-dense, PIM-sparse, RIP, or static.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Routing Policy Match Conditions and Actions on page 392
- Example: Injecting OSPF Routes into the BGP Routing Table on page 402
- Routing Policies Configuration Overview on page 388

Example: Injecting OSPF Routes into the BGP Routing Table

This example shows how to create a policy that injects OSPF routes into the BGP routing table.

- Requirements on page 402
- Overview on page 402
- Configuration on page 402
- Verification on page 404
- Troubleshooting on page 405

Requirements

Before you begin:

- Configure network interfaces.
- Configure external peer sessions. See "Example: Configuring External BGP Point-to-Point Peer Sessions" on page 247.
- Configure interior gateway protocol (IGP) sessions between peers.

Overview

In this example, you create a routing policy called **injectpolicy1** and a routing term called **injectterm1**. The policy injects OSPF routes into the BGP routing table.

Configuration

- Configuring the Routing Policy on page 402
- Configuring Tracing for the Routing Policy on page 404

Configuring the Routing Policy

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Copyright © 2012, Juniper Networks, Inc.

```
set policy-options policy-statement injectpolicy1 term injectterm1 from protocol ospf
set policy-options policy-statement injectpolicy1 term injectterm1 from area 0.0.0.1
set policy-options policy-statement injectpolicy1 term injectterm1 then accept
set protocols bgp export injectpolicy1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To inject OSPF routes into a BGP routing table:

1. Create the policy term.

   ```
   [edit policy-options policy-statement injectpolicy1]
   user@host# set term injectterm1
   ```

2. Specify OSPF as a match condition.

   ```
   [edit policy-options policy-statement injectpolicy1 term injectterm1]
   user@host# set from protocol ospf
   ```

3. Specify the routes from an OSPF area as a match condition.

   ```
   [edit policy-options policy-statement injectpolicy1 term injectterm1]
   user@host# set from area 0.0.0.1
   ```

4. Specify that the route is to be accepted if the previous conditions are matched.

   ```
   [edit policy-options policy-statement injectpolicy1 term injectterm1]
   user@host# set then accept
   ```

5. Apply the routing policy to BGP.

   ```
   [edit]
   user@host# set protocols bgp export injectpolicy1
   ```

Results

Confirm your configuration by entering the **show policy-options** and **show protocols bgp** commands from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show policy-options
policy-statement injectpolicy1 {
   term injectterm1 {
      from {
         protocol ospf;
         area 0.0.0.1;
      }
      then accept;
   }
}

user@host# show protocols bgp
export injectpolicy1;
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Tracing for the Routing Policy

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options policy-statement injectpolicy1 term injectterm1 then trace
set routing-options traceoptions file ospf-bgp-policy-log
set routing-options traceoptions file size 5m
set routing-options traceoptions file files 5
set routing-options traceoptions flag policy
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

1. Include a trace action in the policy.

```
[edit policy-options policy-statement injectpolicy1 term injectterm1]
user@host# then trace
```

2. Configure the tracing file for the output.

```
[edit routing-options traceoptions]
user@host# set file ospf-bgp-policy-log
user@host# set file size 5m
user@host# set file files 5
user@host# set flag policy
```

Results

Confirm your configuration by entering the **show policy-options** and **show routing-options** commands from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show policy-options
policy-statement injectpolicy1 {
   term injectterm1 {
      then {
         trace;
      }
   }
}

user@host# show routing-options
traceoptions {
   file ospf-bgp-policy-log size 5m files 5;
   flag policy;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Copyright © 2012, Juniper Networks, Inc.

Verifying That the Expected BGP Routes Are Present

Purpose Verify the effect of the export policy.

Action From operational mode, enter the **show route** command.

Troubleshooting

- Using the show log Command to Examine the Actions of the Routing Policy on page 405

Using the show log Command to Examine the Actions of the Routing Policy

Problem The routing table contains unexpected routes, or routes are missing from the routing table.

Solution If you configure policy tracing as shown in this example, you can run the **show log ospf-bgp-policy-log** command to diagnose problems with the routing policy. The **show log ospf-bgp-policy-log** command displays information about the routes that the **injectpolicy1** policy term analyzes and acts upon.

Related Documentation

- *Junos OS Policy Framework Configuration Guide*
- Routing Policies Configuration Overview on page 388
- Understanding Protocol-Based Match Conditions on page 402

Autonomous System Path-Based Actions

- Understanding Autonomous System Path-Based Actions on page 405
- Example: Configuring a Routing Policy to Prepend the AS Path on page 406

Understanding Autonomous System Path-Based Actions

You can *prepend* or add one or more autonomous system (AS) numbers at the beginning of an AS path. The AS numbers are added after the local AS number has been added to the path. Prepending an AS path makes a shorter AS path look longer and therefore less preferable to the Border Gateway Protocol (BGP).

For example, from AS 1, there are two equal paths (through AS 2 and AS 3) to reach AS 4. You might want packets from certain sources to use the path through AS 2. Therefore, you must make the path through AS 3 look less preferable so that BGP chooses the path through AS 2. In AS 1, you can prepend multiple AS numbers.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Configuration Overview on page 388
- Understanding Routing Policy Match Conditions and Actions on page 392
- Example: Configuring a Routing Policy to Prepend the AS Path on page 406

Copyright © 2012, Juniper Networks, Inc.

Example: Configuring a Routing Policy to Prepend the AS Path

This example shows how to configure a routing policy to prepend the AS path.

- Requirements on page 406
- Overview on page 406
- Configuration on page 406
- Verification on page 407

Requirements

Before you begin, configure router interfaces and configure routing protocols, as explained in "Routing Policies Configuration Overview" on page 388.

Overview

In this example, you create a routing policy called prependpolicy1 and a term called prependterm1. The routing policy prepends the AS numbers 1111 to routes that are greater than or equal to 172.16.0.0/12, 192.168.0.0/16, and 10.0.0.0/8. The policy is applied as an import policy to all BGP routes and is evaluated when routes are imported to the routing table.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options policy-statement prependpolicy1 term prependterm1 from route-filter
  172.16.0.0/12 orlonger
set policy-options policy-statement prependpolicy1 term prependterm1 from route-filter
  192.168.0.0/16 orlonger
set policy-options policy-statement prependpolicy1 term prependterm1 from route-filter
  10.0.0.0/8 orlonger
set policy-options policy-statement prependpolicy1 term prependterm1 then
  as-path-prepend "1111"
set policy-options policy-statement test term 1 from protocol direct
set protocols bgp import prependpolicy1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To create a routing policy that prepends AS numbers to multiple routes:

1. Create the routing policy.

   ```
   [edit]
   user@host# edit policy-options policy-statement prependpolicy1
   ```

2. Create the routing term.

   ```
   [edit policy-options policy-statement prependpolicy1]
   user@host# edit term prependterm1
   ```


Copyright © 2012, Juniper Networks, Inc.

3. Specify the routes to prepend with AS numbers.

```
[edit policy-options policy-statement prependpolicy1 term prependterm1]
user@host# set from route-filter 172.16.0.0/12 orlonger
user@host# set from route-filter 192.168.0.0/16 orlonger
user@host# set from route-filter 10.0.0.0/8 orlonger
```

4. Specify the AS numbers to prepend.

```
[edit policy-options policy-statement prependpolicy1 term prependterm1]
user@host# set then as-path-prepend “1 1 1 1”
```

NOTE: If you enter multiple numbers, you must separate each number with a space. Enclose the numbers in double quotation marks.

5. Apply the policy as an import policy for all BGP routes.

```
[edit]
user@host# set protocols bgp import prependpolicy1
```

NOTE: You can refer to the same routing policy one or more times in the same or different import statement.

Results Confirm your configuration by entering the **show policy-options** and **show protocols bgp** commands from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show policy-options
policy-statement prependpolicy1 {
   term prependterm1 {
      from {
         route-filter 172.16.0.0/12 orlonger;
         route-filter 192.168.0.0/16 orlonger;
         route-filter 10.0.0.0/8 orlonger;
      }
      then as-path-prepend "1 1 1 1";
   }
}

user@host# show protocols bgp
import prependpolicy1;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the AS Numbers to Prepend on page 408
- Verifying the Routing Policy on page 408

Verifying the AS Numbers to Prepend

Purpose Verify that the policy and term are configured on the device and that the appropriate routes are specified to prepend with AS numbers.

Action From operational mode, enter the **show policy-options** command.

Verifying the Routing Policy

Purpose Verify that the routing policy is applied to the routing protocol.

Action From operational mode, enter the **show protocols bgp** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Configuration Overview on page 388
- Understanding Autonomous System Path-Based Actions on page 405

Routing Policy Damping Parameters

- Understanding Damping Parameters on page 408
- Example: Configuring Damping Parameters on page 409

Understanding Damping Parameters

BGP *route flapping* describes the situation in which BGP systems send an excessive number of update messages to advertise network reachability information. BGP *flap damping* is a 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.

Flap damping reduces the number of update messages by marking routes as ineligible for selection as the active or preferable route. Marking routes in this way leads to some delay, or *suppression*, in the propagation of route information, but the result is increased network stability. You typically apply flap damping to external BGP (EBGP) routes (routes in different ASs). You can also apply flap damping within a confederation, between confederation member ASs. Because routing consistency within an AS is important, do not apply flap damping to internal BGP (IBGP) routes. (If you do, it is ignored.)

By default, route flap damping is not enabled. Damping is applied to external peers and to peers at confederation boundaries.

When you enable damping, default parameters are applied, as summarized in Table 13 on page 409.

 Copyright © 2012, Juniper Networks, Inc.

Table 13: Damping Parameters

Damping Parameter	Description	Default Value	Possible Values
half-life ***minutes***	Decay half-life—Number of minutes after which an arbitrary value is halved if a route stays stable.	**15** (minutes)	**1** through **4**
max-suppress ***minutes***	Maximum hold-down time for a route, in minutes.	**60** (minutes)	**1** through **720**
reuse	Reuse threshold—Arbitrary value below which a suppressed route can be used again.	**750**	**1** through **20,000**
suppress	Cutoff (suppression) threshold—Arbitrary value above which a route can no longer be used or included in advertisements.	**3000**	**1** through **20,000**

To change the default BGP flap damping values, you define actions by creating a named set of damping parameters and including it in a routing policy with the damping action. For the damping routing policy to work, you also must enable BGP route flap damping.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Overview on page 387
- Example: Configuring Damping Parameters on page 409

Example: Configuring Damping Parameters

This example shows how to configure damping parameters.

- Requirements on page 409
- Overview on page 409
- Configuration on page 410
- Verification on page 412

Requirements

Before you begin, configure router interfaces and configure routing protocols, as explained in "Routing Policies Configuration Overview" on page 388.

Overview

In this example, you configure a routing policy called policy1 and a corresponding routing term called term1. Within the term, you configure the route filter to include source routes greater than or equal to 10.210.0.0/16 and destination routes greater than or equal to 10.215.0.0/16. Then you group the source and destination prefixes into a forwarding class called forwarding-class1 and apply policy1 to the forwarding table. The routing policy is evaluated when routes are being exported from the routing table into the forwarding table. Only the active routes are exported from the routing table.

Copyright © 2012, Juniper Networks, Inc.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options policy-statement dampenpolicy1 term dampenterm1 from route-filter
  172.16.0.0/12 orlonger damping group1
set policy-options policy-statement dampenpolicy1 term dampenterm1 from route-filter
  192.168.0.0/16 orlonger
set policy-options policy-statement dampenpolicy1 term dampenterm1 from route-filter
  10.0.0.0/8 orlonger
set policy-options policy-statement test term 1 from protocol direct
set policy-options damping group1 half-life 30
set policy-options damping group1 reuse 750
set policy-options damping group1 suppress 3000
set policy-options damping group1 max-suppress 60
set policy-options damping group2 half-life 40
set policy-options damping group2 reuse 1000
set policy-options damping group2 suppress 400
set policy-options damping group2 max-suppress 45
set policy-options damping group3 disable
set protocols bgp damping
set protocols bgp group groupA neighbor 172.16.15.14 import dampenpolicy1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure damping parameters:

1. Specify the routes to dampen and associate each group of routes with a group name.

   ```
   [edit policy-options policy-statement dampenpolicy1 term dampenterm1]
   user@host# set from route-filter 172.16.0.0/12 orlonger damping group1
   user@host# set from route-filter 192.168.0.0/16 orlonger
   user@host# set from route-filter 10.0.0.0/8 orlonger
   ```

2. Create and configure the damping parameter groups.

   ```
   [edit policy-options damping]
   user@host# set group1 half-life 30 max-suppress 60 reuse 750 suppress 3000
   user@host# set group2 half-life 40 max-suppress 45 reuse 1000 suppress 400
   user@host# set group3 disable
   ```

3. Enable damping for BGP.

   ```
   [edit]
   user@host# set protocols bgp damping
   ```

4. Apply the policy as an import policy for the BGP neighbor.

   ```
   [edit ]
   ```


Copyright © 2012, Juniper Networks, Inc.

```
user@host# set protocols bgp group groupA neighbor 172.16.15.14 import
  dampenpolicy1
```

NOTE: You can refer to the same routing policy one or more times in the same or different import statement.

Results Confirm your configuration by entering the **show policy-options** and **show protocols bgp** commands from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show policy-options
policy-statement dampenpolicy1 {
  term dampenterm1 {
    from {
      route-filter 172.16.0.0/12 orlonger damping group1;
      route-filter 192.168.0.0/16 orlonger;
      route-filter 10.0.0.0/8 orlonger;
    }
  }
}
  damping group1 {
    half-life 30;
    reuse 750;
    suppress 3000;
    max-suppress 60;
  }
  damping group2 {
    half-life 40;
    reuse 1000;
    suppress 400;
    max-suppress 45;
  }
  damping group3 {
    disable;
  }
```

```
user@host# show protocols bgp
damping;
group groupA {
  neighbor 172.16.15.14 {
    import dampenpolicy1;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying the Damping Parameters on page 412
- Verifying the Routing Policy on page 412

Verifying the Damping Parameters

Purpose Verify that the policy and term are configured on the device and that the appropriate damping parameters are specified within the term.

Action From operational mode, enter the **show policy-options** command.

Verifying the Routing Policy

Purpose Verify that damping is enabled for BGP and that the routing policy is applied to the routing protocol.

Action From operational mode, enter the **show protocols bgp** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Routing Policies Configuration Overview on page 388
- Understanding Damping Parameters on page 408

ECMP Flow-Based Forwarding

- Understanding ECMP Flow-Based Forwarding on page 412
- Example: Configuring ECMP Flow-Based Forwarding on page 413

Understanding ECMP Flow-Based Forwarding

An equal-cost multipath (ECMP) set is formed when the routing table contains multiple next-hop addresses for the same destination with equal cost. (Routes of equal cost have the same preference and metric values.) If there is an ECMP set for the active route, the Junos OS software uses a hash algorithm to choose *one* of the next-hop addresses in the ECMP set to install in the forwarding table.

You can configure Junos OS so that multiple next-hop entries in an ECMP set are installed in the forwarding table. On Juniper Networks devices, per-flow load balancing can be performed to spread traffic across multiple paths between routing devices. On Juniper Networks security devices, source and destination IP addresses and protocols are examined to determine individual traffic flows. Packets for the same flow are forwarded on the same interface; the interface does not change when there are additions or changes to the ECMP set. This is important for features such as source NAT, where the translation is performed only during the first path of session establishment; IDP; ALG; and route-based VPN tunnels. If a packet arrives on a given interface in an ECMP set, the security device ensures that reverse traffic is forwarded through the same interface.

 Copyright © 2012, Juniper Networks, Inc.

NOTE: ECMP flow-based forwarding on security devices applies only to IPv4 unicast traffic flows. Multicast and IPv6 flows are not supported.

On Juniper Networks security devices, the maximum number of next-hop addresses in an ECMP set that can be installed in the forwarding table is 16. If there are more than 16 next-hop addresses in an ECMP set, only the first 16 addresses are used.

In a chassis cluster deployment, a *local* interface is an interface that is on the same node as the interface on which a packet arrives, and a *remote* interface is an interface that is on the other chassis cluster node. If an ECMP route has both local and remote interfaces in a chassis cluster, then the local interface is favored for the next hop.

If a next-hop address is no longer part of the ECMP set or if it is removed from the routing table because of a route change, a flow that uses the next hop is rerouted and the session is not affected. Rerouting of the flow also occurs if there is a configuration change that takes away the next-hop address or if an administrator takes down the next-hop interface without deleting it. If a next-hop address is removed from the routing table because the interface is deleted or the session is intentionally cleared, the session is killed without being rerouted.

NOTE: We recommend that interfaces in an ECMP set be in the same security zone. If a flow is rerouted and the rerouted flow uses an interface in a different security zone than the original route, the session is killed.

To configure ECMP flow-based forwarding on Juniper Networks security devices, first define a load-balancing routing policy by including one or more **policy-statement** configuration statements at the [**edit policy-options**] hierarchy level, with the action **load-balance per-packet**. Then apply the routing policy to routes exported from the routing table to the forwarding table. To do this, include the **forwarding-table** and **export** configuration statements at the [**edit routing-options**] hierarchy level.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring ECMP Flow-Based Forwarding on page 413
- Routing Policies Overview on page 387
- Understanding Routing Policy Match Conditions and Actions on page 392

Example: Configuring ECMP Flow-Based Forwarding

This example shows how to configure ECMP flow-based forwarding.

- Requirements on page 414
- Overview on page 414
- Configuration on page 414
- Verification on page 417

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

This example configures three static ECMP routes on an SRX Series device. Each static route uses a different router to reach the destination server. The interfaces to the routers are assigned to the untrust security zone. This example creates a load-balancing routing policy named **load-balancing-policy** and applies the policy to all routes exported from the routing table to the forwarding table.

Topology

Figure 54 on page 414 shows the topology used in this example.

Figure 54: ECMP Routes

Configuration

CLI Quick Configuration

To quickly configure ECMP flow-based forwarding, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your

Copyright © 2012, Juniper Networks, Inc.

network configuration, and then copy and paste the commands into the CLI at the [**edit**] hierarchy level.

```
set interfaces ge-0/0/2 unit 0 family inet address 22.0.39.56/8
set interfaces ge-0/0/4 unit 0 family inet address 23.0.39.56/8
set interfaces ge-0/0/6 unit 0 family inet address 24.0.39.56/8
set interfaces ge-0/0/7 unit 0 family inet address 25.0.39.56/8
set routing-options static route 26.0.0.0/8 next-hop 23.0.54.111
set routing-options static route 26.0.0.0/8 next-hop 24.0.44.101
set routing-options static route 26.0.0.0/8 next-hop 25.0.44.106
set security zones security-zone trust interfaces ge-0/0/2
set security zones security-zone untrust interfaces ge-0/0/4 ge-0/0/6 ge-0/0/7
set security policies from-zone trust to-zone untrust policy permit-mail match
  source-address 22.0.39.56/8
set security policies from-zone trust to-zone untrust policy permit-mail match
  destination-address 26.0.0.0/8
set security policies from-zone trust to-zone untrust policy permit-mail match application
  junos-mail
set security policies from-zone trust to-zone untrust policy permit-mail then permit
set policy-options policy-statement load-balancing-policy then load-balance per-packet
set routing-options forwarding-table export load-balancing-policy
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure ECMP flow-based forwarding:

1. Configure interfaces.

```
[edit interaces]
user@host# set ge-0/0/2 unit 0 family inet address 22.0.39.56/8
user@host# set ge-0/0/4 unit 0 family inet address 23.0.39.56/8
user@host# set ge-0/0/6 unit 0 family inet address 24.0.39.56/8
user@host# set ge-0/0/7 unit 0 family inet address 25.0.39.56/8
```

2. Create security zones.

```
[edit security]
user@host# set zones security-zone trust interfaces ge-0/0/2
user@host# set zones security-zone untrust interfaces ge-0/0/4
user@host# set zones security-zone untrust interfaces ge-0/0/6
user@host# set zones security-zone untrust interfaces ge-0/0/7
```

3. Configure a security policy.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy permit-mail match source-address 22.0.39.56/8
user@host# set policy permit-mail match destination-address 26.0.0.0/8
user@host# set policy permit-mail match application junos-mail
user@host# set policy permit-mail then permit
```

4. Configure ECMP routes.

```
[edit routing-options]
user@host# set static route 26.0.0.0/8 next-hop 23.0.54.111
user@host# set static route 26.0.0.0/8 next-hop 24.0.44.101
user@host# set static route 26.0.0.0/8 next-hop 25.0.44.106
```

5. Create a load-balancing routing policy.

```
[edit policy-options]
user@host# set policy-statement load-balancing-policy then load-balance per-packet
```

6. Apply the routing policy to all routes exported from the routing table to the forwarding table.

```
[edit routing-options]
user@host# set forwarding-table export load-balancing-policy
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show security**, **show policy-options** and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/2 {
   unit 0 {
      family inet {
         address 22.0.39.56/8;
      }
   }
}
ge-0/0/4 {
   unit 0 {
      family inet {
         address 23.0.39.56/8;
      }
   }
}
ge-0/0/6 {
   unit 0 {
      family inet {
         address 24.0.39.56/8;
      }
   }
}
ge-0/0/7 {
   unit 0 {
      family inet {
         address 25.0.39.56/8;
      }
   }
}
user@host# show security
policies {
   from-zone trust to-zone untrust {
      policy permit-mail {
         match {
            source-address 22.0.39.56/8;
            destination-address 26.0.0.0/8;
            application junos-mail;
         }
         then {
```

 Copyright © 2012, Juniper Networks, Inc.

```
            permit;
          }
        }
      }
    }
    zones {
      security-zone trust {
        interfaces {
          ge-0/0/2.0;
        }
      }
      security-zone untrust {
        interfaces {
          ge-0/0/4.0;
          ge-0/0/6.0;
          ge-0/0/7.0;
        }
      }
    }
    user@host# show policy-options
    policy-statement load-balancing-policy {
      then {
        load-balance per-packet;
      }
    }

    [edit]
    user@host# show routing-options
    forwarding-table {
      export load-balancing-policy;
    }
    static {
      route 0.0.0.0/0 next-hop 10.100.37.1;
      route 26.0.0.0/8 next-hop [ 23.0.54.111 25.0.44.106 24.0.44.101 ];
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying Forwarding Table

Purpose Verify that the route information for all ECMP routes appears in the forwarding table.

Action From operational mode, enter the **show route forwarding-table** command.

```
user@host> show route forwarding-table
  Routing table: default.inet
  Internet:
  Destination      Type RtRef Next hop        Type Index NhRef Netif
  ...

  26.0.0.0/8       user    0 23.0.54.111      rslv    0    1 ge-0/0/4.0
  26.0.0.0/8       user    0 24.0.44.101      rslv    0    1 ge-0/0/6.0
  26.0.0.0/8       user    0 25.0.44.106      rslv    0    1 ge-0/0/7.0
```

...

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding ECMP Flow-Based Forwarding on page 412

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 15

Stateless Firewall Filters

- Introduction to Stateless Firewall Filters on page 419
- Guidelines for Configuring Standard Firewall Filters on page 424
- Guidelines for Applying Standard Firewall Filters on page 429
- Stateless Firewall Filter Terms on page 431
- Trusted Source and Flood Prevention Stateless Firewall Filters on page 475
- Fragment Handling Stateless Firewall Filters on page 488

Introduction to Stateless Firewall Filters

- Router Data Flow Overview on page 419
- Stateless Firewall Filter Overview on page 421
- Standard Stateless Firewall Filter Overview on page 422
- Stateless Firewall Filter Types on page 423

Router Data Flow Overview

The Junos OS provides a *policy framework*, which is a collection of Junos OS policies that enable you to control flows of routing information and packets within the router.

- Flow of Routing Information on page 419
- Flow of Data Packets on page 420
- Flow of Local Packets on page 420
- Interdependent Flows of Routing Information and Packets on page 420

Flow of Routing Information

Routing information is the information about routes learned by the routing protocols from a router's neighbors. This information is stored in routing tables. The routing protocols advertise active routes only from the routing tables. An *active route* is a route that is chosen from all routes in the routing table to reach a destination.

To control which routes the routing protocols place in the routing tables and which routes the routing protocols advertise from the routing tables, you can configure *routing policies*, which are sets of rules that the policy framework uses to preempt default routing policies.

Copyright © 2012, Juniper Networks, Inc.

The Routing Engine, which is the router's control plane, handles the flow of routing information between the routing protocols and the routing tables and between the routing tables and the forwarding table. The Routing Engine runs the Junos OS and routing policies and stores the active router configuration, the master routing table, and the master forwarding table,

Flow of Data Packets

Data packets are chunks of data that transit the router as they are being forwarded from a source to a destination. When a router receives a data packet on an interface, it determines where to forward the packet by looking in the forwarding table for the best route to a destination. The router then forwards the data packet toward its destination through the appropriate interface.

The Packet Forwarding Engine, which is the router's forwarding plane, handles the flow of data packets in and out of the router's physical interfaces. Although the Packet Forwarding Engine contains Layer 3 and Layer 4 header information, it does not contain the packet data itself (the packet's payload).

Flow of Local Packets

Local packets are chunks of data that are destined for or sent by the router. Local packets usually contain routing protocol data, data for IP services such as Telnet or SSH, and data for administrative protocols such as the Internet Control Message Protocol (ICMP). When the Routing Engine receives a local packet, it forwards the packet to the appropriate process or to the kernel, which are both part of the Routing Engine, or to the Packet Forwarding Engine.

The Routing Engine handles the flow of local packets from the router's physical interfaces and to the Routing Engine.

Interdependent Flows of Routing Information and Packets

Figure 55 on page 420 illustrates the flow of data through a router. Although routing information flows and packet flows are very different from one another, they are also interdependent.

Figure 55: Flows of Routing Information and Packets

Routing policies determine which routes the Routing Engine places in the forwarding table. The forwarding table, in turn, has an integral role in determining the appropriate physical interface through which to forward a packet.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Stateless Firewall Filter Overview on page 421
- "Packet Flow Within Routers Overview" in the *Junos OS Class of Service Configuration Guide*
- "Understanding BGP Path Selection on page 265" in the *Junos OS Routing Protocols Configuration Guide*
- "Understanding Route Preference Values" in the *Junos OS Routing Protocols Configuration Guide*
- "Routing Policy Overview" in the *Junos OS Policy Framework Configuration Guide*

Stateless Firewall Filter Overview

This topic covers the following information:

- Packet Flow Control on page 421
- Stateless and Stateful Firewall Filters on page 421
- Purpose of Stateless Firewall Filters on page 422

Packet Flow Control

To influence which packets are allowed to transit the system and to apply special actions to packets as necessary, you can configure *stateless firewall filters*. A stateless firewall specifies a sequence of one or more packet-filtering rules, called *filter terms*. A filter term specifies *match conditions* to use to determine a match and *actions* to take on a matched packet. A stateless firewall filter enables you to manipulate any packet of a particular protocol family, including fragmented packets, based on evaluation of Layer 3 and Layer 4 header fields. You typically apply a stateless firewall filter to one or more interfaces that have been configured with protocol family features. You can apply a stateless firewall filter to an ingress interface, an egress interface, or both.

Data Packet Flow Control

To control the flow of data packets transiting the device as the packets are being forwarded from a source to a destination, you can apply stateless firewall filters to the input or output of the router's physical interfaces.

To enforce a specified bandwidth and maximum burst size for traffic sent or received on an interface, you can configure *policers*. Policers are a specialized type of stateless firewall filter and a primary component of the Junos OS *class-of-service* (CoS).

Local Packet Flow Control

To control the flow of local packets between the physical interfaces and the Routing Engine, you can apply stateless firewall filters to the input or output of the *loopback interface*. The loopback interface (**lo0**) is the interface to the Routing Engine and carries no data packets.

Stateless and Stateful Firewall Filters

A stateless firewall filter, also known as an *access control list* (ACL), does not statefully inspect traffic. Instead, it evaluates packet contents statically and does not keep track

of the state of network connections. In contrast, a *stateful firewall filter* uses connection state information derived from other applications and past communications in the data flow to make dynamic control decisions.

The *Junos OS Firewall Filter and Policer Configuration Guide* describes *stateless firewall filters* supported on T Series, M Series, and MX Series routers. For information about Junos OS stateful firewall policies for J Series and SRX Series security devices, see "Security Policies Overview" in the *Junos OS Security Configuration Guide*.

Purpose of Stateless Firewall Filters

The basic purpose of a stateless firewall filter is to enhance security through the use of packet filtering. Packet filtering enables you to inspect the components of incoming or outgoing packets and then perform the actions you specify on packets that match the criteria you specify. The typical use of a stateless firewall filter is to protect the Routing Engine processes and resources from malicious or untrusted packets.

Related Documentation

- Router Data Flow Overview on page 419
- Stateless Firewall Filter Types on page 423
- "Traffic Policing Overview" in the *Junos OS Firewall Filter and Policer Configuration Guide*
- "Packet Flow Through the CoS Process Overview" in the *Junos OS Class of Service Configuration Guide*

Standard Stateless Firewall Filter Overview

Firewall filters provide a means of protecting your router from excessive traffic transiting the router to a network destination or destined for the Routing Engine. Firewall filters that control local packets can also protect your router from external incidents.

You can configure a firewall filter to do the following:

- Restrict traffic destined for the Routing Engine based on its source, protocol, and application.
- Limit the traffic rate of packets destined for the Routing Engine to protect against flood, or denial-of-service (DoS) attacks.
- Address special circumstances associated with fragmented packets destined for the Routing Engine. Because the device evaluates every packet against a firewall filter (including fragments), you must configure the filter to accommodate fragments that do not contain packet header information. Otherwise, the filter discards all but the first fragment of a fragmented packet.

Related Documentation

- Stateless Firewall Filter Types on page 423
- Guidelines for Configuring Standard Firewall Filters on page 424
- Guidelines for Applying Standard Firewall Filters on page 429
- Understanding How to Use Standard Firewall Filters on page 475

 Copyright © 2012, Juniper Networks, Inc.

Stateless Firewall Filter Types

This topic covers the following information:

- Standard Stateless Firewall Filters on page 423
- Service Filters on page 423
- Simple Filters on page 423

Standard Stateless Firewall Filters

The Junos OS standard stateless firewall filters support a rich set of packet-matching criteria that you can use to match on specific traffic and perform specific actions, such as forwarding or dropping packets that match the criteria you specify. You can configure firewall filters to protect the local router or to protect another device that is either directly or indirectly connected to the local router. For example, you can use the filters to restrict the local packets that pass from the router's physical interfaces to the Routing Engine. Such filters are useful in protecting the IP services that run on the Routing Engine, such as Telnet, SSH, and BGP, from denial-of-service attacks.

NOTE: If you configured targeted broadcast for virtual routing and forwarding (VRF) by including the forward-and-send-to-re statement, any firewall filter that is configured on the Routing Engine loopback interface (lo0) cannot be applied to the targeted broadcast packets that are forwarded to the Routing Engine. This is because broadcast packets are forwarded as flood next hop traffic and not as local next hop traffic, and you can only apply a firewall filter to local next hop routes for traffic directed toward the Routing Engine.

Service Filters

A service filter defines packet-filtering (a set of match conditions and a set of actions) for IPv4 or IPv6 traffic. You can apply a service filter to the inbound or outbound traffic at an adaptive services interface to perform packet filtering on traffic before it is accepted for service processing. You can also apply a service filter to the traffic that is returning to the services interface after service processing to perform postservice processing.

Service filters filter IPv4 and IPv6 traffic only and can be applied to logical interfaces on Adaptive Services PICs, MultiServices PICs, and MultiServices DPCs only. Service filters are not supported on J Series devices and Branch SRX devices.

Simple Filters

Simple filters are supported on Gigabit Ethernet intelligent queuing (IQ2) and Enhanced Queuing Dense Port Concentrator (EQ DPC) interfaces only. Unlike standard filters, simple filters support IPv4 traffic only and have a number of restrictions. For example, you cannot configure a terminating action for a simple filter. Simple filters always accept packets. Also, simple filters can be applied only as input filters. They are not supported on outbound traffic. Simple filters are recommended for metropolitan Ethernet applications.

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Stateless Firewall Filter Overview on page 421
- Stateless Firewall Filter Components on page 431

Guidelines for Configuring Standard Firewall Filters

This topic covers the following information:

- Statement Hierarchy for Configuring Standard Firewall Filters on page 424
- Standard Firewall Filter Protocol Families on page 425
- Standard Firewall Filter Names and Options on page 425
- Standard Firewall Filter Terms on page 425
- Standard Firewall Filter Match Conditions on page 426
- Standard Firewall Filter Actions on page 427

Statement Hierarchy for Configuring Standard Firewall Filters

To configure a standard firewall filter, you can include the following statements. For an IPv4 standard firewall filter, the **family inet** statement is optional.

```
firewall {
  family family-name {
    filter filter-name {
      accounting-profile name;
      interface-specific;
      physical-interface-filter;
      term term-name {
        filter filter-name;
      }
      term term-name {
        from {
          match-conditions;
          ip-version ipv4 {
            match-conditions;
            protocol (tcp | udp) {
              match conditions;
            }
          }
        }
        then {
          actions;
        }
      }
    }
  }
}
```

You can include the firewall configuration at one of the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

Copyright © 2012, Juniper Networks, Inc.

NOTE: For stateless firewall filtering, you must allow the output tunnel traffic through the firewall filter applied to input traffic on the interface that is the next-hop interface toward the tunnel destination. The firewall filter affects only the packets exiting the router by way of the tunnel.

Standard Firewall Filter Protocol Families

A standard firewall filter configuration is specific to a particular protocol family. Under the **firewall** statement, include one of the following statements to specify the protocol family for which you want to filter traffic:

- **family any**—To filter protocol-independent traffic.
- **family inet**—To filter Internet Protocol version 4 (IPv4) traffic.
- **family inet6**—To filter Internet Protocol version 6 (IPv6) traffic.
- **family mpls**—To filter MPLS traffic.
- **family vpls**—To filter virtual private LAN service (VPLS) traffic.
- **family ccc**—To filter Layer 2 circuit cross-connection (CCC) traffic.
- **family bridge**—To filter Layer 2 bridging traffic for MX Series 3D Universal Edge Routers only.

The **family *family-name*** statement is required only to specify a protocol family other than IPv4. To configure an IPv4 firewall filter, you can configure the filter at the **[edit firewall]** hierarchy level without including the **family inet** statement, because the **[edit firewall]** and **[edit firewall family inet]** hierarchy levels are equivalent.

Standard Firewall Filter Names and Options

Under the **family *family-name*** statement, you can include **filter *filter-name*** statements to create and name standard firewall filters. The filter name can contain letters, numbers, and hyphens (-) and be up to 64 characters long. To include spaces in the name, enclose the entire name in quotation marks (“ ”).

At the **[edit firewall family *family-name* filter *filter-name*]** hierarchy level, the following statements are optional:

- **accounting-profile**
- **interface-specific**
- **physical-interface-filter**

Standard Firewall Filter Terms

Under the **filter *filter-name*** statement, you can include **term *term-name*** statements to create and name filter terms.

- You must configure at least one term in a firewall filter.

Copyright © 2012, Juniper Networks, Inc.

- You must specify a unique name for each term within a firewall filter. The term name can contain letters, numbers, and hyphens (-) and can be up to 64 characters long. To include spaces in the name, enclose the entire name in quotation marks (“ ”).
- The order in which you specify terms within a firewall filter configuration is important. Firewall filter terms are evaluated in the order in which they are configured. By default, new terms are always added to the end of the existing filter. You can use the **insert** configuration mode command to reorder the terms of a firewall filter.

At the **[edit firewall family *family-name* filter *filter-name* term *term-name*]** hierarchy level, the **filter *filter-name*** statement is not valid in the same term as **from** or **then** statements. When included at this hierarchy level, the **filter *filter-name*** statement is used to *nest* firewall filters.

Standard Firewall Filter Match Conditions

Standard firewall filter match conditions are specific to the type of traffic being filtered.

With the exception of MPLS-tagged IPv4 traffic, you specify the term’s match conditions under the **from** statement. For MPLS-tagged IPv4 traffic, you specify the term’s IPv4 address-specific match conditions under the **ip-version ipv4** statement and the term’s IPv4 port-specific match conditions under the **protocol (tcp | udp)** statement.

Table 14 on page 426 describes the types of traffic for which you can configure standard stateless firewall filters.

Table 14: Standard Firewall Filter Match Conditions by Protocol Family

Traffic Type	Hierarchy Level at Which Match Conditions Are Specified
Protocol-independent	**[edit firewall family any filter *filter-name* term *term-name*]** For the complete list of match conditions, see Standard Firewall Filter Match Conditions for Protocol-Independent Traffic.
IPv4	**[edit firewall family inet filter *filter-name* term *term-name*]** For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for IPv4 Traffic” on page 436.
IPv6	**[edit firewall family inet6 filter *filter-name* term *term-name*]** For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for IPv6 Traffic” on page 446.
MPLS	**[edit firewall family mpls filter *filter-name* term *term-name*]** For the complete list of match conditions, see Standard Firewall Filter Match Conditions for MPLS Traffic.
IPv4 addresses in MPLS flows	**[edit firewall family mpls filter *filter-name* term *term-name* ip-version ipv4]** For the complete list of match conditions, see Standard Firewall Filter Match Conditions for MPLS-Tagged IPv4 Traffic.

Copyright © 2012, Juniper Networks, Inc.

Table 14: Standard Firewall Filter Match Conditions by Protocol Family *(continued)*

Traffic Type	Hierarchy Level at Which Match Conditions Are Specified
IPv4 ports in MPLS flows	**[edit firewall family mpls filter *filter-name* term *term-name* ip-version ipv4 protocol (tcp | udp)]** For the complete list of match conditions, see Standard Firewall Filter Match Conditions for MPLS-Tagged IPv4 Traffic.
VPLS	**[edit firewall family vpls filter *filter-name* term *term-name*]** For the complete list of match conditions, see Standard Firewall Filter Match Conditions for VPLS Traffic.
Layer 2 CCC	**[edit firewall family ccc filter *filter-name* term *term-name*]** For the complete list of match conditions, see Standard Firewall Filter Match Conditions for Layer 2 CCC Traffic.
Layer 2 Bridging (MX Series routers only)	**[edit firewall family bridge filter *filter-name* term *term-name*]** For the complete list of match conditions, see Standard Firewall Filter Match Conditions for Layer 2 Bridging Traffic.

If you specify an IPv6 address in a match condition (the **address**, **destination-address**, or **source-address** match conditions), use the syntax for text representations described in RFC 2373, *IP Version 6 Addressing Architecture*. For more information about IPv6 addresses, see "IPv6 Overview" and "IPv6 Standards" in the *Junos OS Routing Protocols Configuration Guide*.

Standard Firewall Filter Actions

Under the **then** statement for a standard stateless firewall filter term, you can specify the actions to be taken on a packet that matches the term.

Table 15 on page 428 summarizes the types of actions you can specify in a standard stateless firewall filter term.

Table 15: Standard Firewall Filter Action Categories

Type of Action	Description	Comment
Terminating	Halts all evaluation of a firewall filter for a specific packet. The router performs the specified action, and no additional terms are used to examine the packet. You can specify only one *terminating action* in a standard firewall filter. You can, however, specify one terminating action with one or more *nonterminating actions* in a single term. For example, within a term, you can specify **accept** with **count** and **syslog**.	See "Standard Firewall Filter Terminating Actions" on page 468.
Nonterminating	Performs other functions on a packet (such as incrementing a counter, logging information about the packet header, sampling the packet data, or sending information to a remote host using the system log functionality), but any additional terms are used to examine the packet.	See "Standard Firewall Filter Nonterminating Actions" on page 469.
Flow control	For standard stateless firewall filters only, the **next term** action directs the router to perform configured actions on the packet and then, rather than terminate the filter, use the next term in the filter to evaluate the packet. If the **next term** action is included, the matching packet is evaluated against the next term in the firewall filter. Otherwise, the matching packet is not evaluated against subsequent terms in the firewall filter. For example, when you configure a term with the nonterminating action **count**, the term's action changes from an implicit **discard** to an implicit **accept**. The **next term** action forces the continued evaluation of the firewall filter.	You cannot configure the **next term** action with a terminating action in the same filter term. However, you can configure the next term action with another nonterminating action in the same filter term. A maximum of 1024 **next term** actions are supported per standard stateless firewall filter configuration. If you configure a standard filter that exceeds this limit, your candidate configuration results in a commit error.

Related Documentation

- Guidelines for Applying Standard Firewall Filters on page 429
- Understanding How to Use Standard Firewall Filters on page 475

Copyright © 2012, Juniper Networks, Inc.

Guidelines for Applying Standard Firewall Filters

This topic covers the following information:

- Applying Standard Firewall Filters Overview on page 429
- Statement Hierarchy for Applying Standard Firewall Filters on page 429
- Restrictions on Applying Standard Firewall Filters on page 430

Applying Standard Firewall Filters Overview

You can apply a standard stateless firewall filter to a physical interface on the router or to the loopback interface on the router. You can apply a firewall filter to a single interface or to multiple interfaces on the router.

Applying a Firewall Filter to a Router's Physical Interfaces

When you apply a standard firewall filter to a physical interface on the router, the filter evaluates all data packet that pass through that interface.

Applying a Firewall Filter to the Router's Loopback Interface

The router's loopback interface, **lo0**, is the interface to the Routing Engine and carries no data packets. When you apply a standard firewall filter to the loopback interface, the filter evaluates the local packets received or transmitted by the Routing Engine.

Applying a Firewall Filter to Multiple Interfaces

You can use the same standard firewall filter one or more times.

On M Series routers, except the M120 and M320 routers, if you apply a firewall filter to multiple interfaces, the filter acts on the sum of traffic entering or exiting those interfaces.

On T Series, M120, M320, and MX Series routers, interfaces are distributed among multiple packet- forwarding components. On these routers, you can configure standard stateless firewall filters and service filters that, when applied to multiple interfaces, act on the individual traffic streams entering or exiting each interface, regardless of the sum of traffic on the multiple interfaces.

For more information, see Interface-Specific Firewall Filter Instances Overview.

Statement Hierarchy for Applying Standard Firewall Filters

To apply a standard stateless firewall filter to a logical interface, configure the **input** ***filter-name***, **input-list** ***filter-name***, **output** ***filter-name***, or **output-list** ***filter-name*** statements in the **filter** stanza for the logical interface protocol family.

```
interfaces {
  interface-name {
    unit logical-unit-number {
      family family-name {
        ...
        filter {
          group group-number;
```

```
                input filter-name;
                input-list [ filter-names ];
                output filter-name;
                output-list [ filter-names ];
            }
        }
    }
  }
}
```

You can include the interface configuration at one of the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

Restrictions on Applying Standard Firewall Filters

- Number of Input and Output Filters Per Logical Interface on page 430
- MPLS and Layer 2 CCC Firewall Filters in Lists on page 430
- Layer 2 CCC Firewall Filters on MX Series Routers on page 431
- Protocol-Independent Firewall Filters on the Loopback Interface on page 431

Number of Input and Output Filters Per Logical Interface

Input filters—Although you can use the same filter multiple times, you can apply only one input filter or one input filter list to an interface.

- To specify a single firewall filter to be used to evaluate packets received on the interface, include the **input *filter-name*** statement in the **filter** stanza.
- To specify an ordered list of firewall filters to be used to evaluate packets received on the interface, include the **input-list [*filter-names*]** statement in the **filter** stanza. You can specify up to 16 firewall filters for the filter input list.

Output filters—Although you can use the same filter multiple times, you can apply only one output filter or one output filter list to an interface.

- To specify a single firewall filter to be used to evaluate packets transmitted on the interface, include the **output *filter-name*** statement in the **filter** stanza.
- To specify an ordered list of firewall filters to be used to evaluate packets transmitted on the interface, include the **output-list [*filter-names*]** statement in the **filter** stanza. You can specify up to 16 firewall filters in a filter output list.

MPLS and Layer 2 CCC Firewall Filters in Lists

The **input-list *filter-names*** and **output-list *filter-names*** statements for firewall filters for the **ccc** and **mpls** protocol families are supported on all interfaces with the exception of the following:

- Management interfaces and internal Ethernet interfaces (**fxp** or **em0**)
- Loopback interfaces (**lo0**)

 Copyright © 2012, Juniper Networks, Inc.

- USB modem interfaces (**umd**)

Layer 2 CCC Firewall Filters on MX Series Routers

On MX Series routers only, you cannot apply a Layer 2 CCC stateless firewall filter (a firewall filter configured at the **[edit firewall filter family ccc]** hierarchy level) as an output filter. On MX Series routers, firewall filters configured for the **family ccc** statement can be applied only as input filters.

Protocol-Independent Firewall Filters on the Loopback Interface

Protocol-independent firewall filters—stateless firewall filters configured at the **[edit firewall family any]** hierarchy level— are not supported on the router loopback interface (**lo0**).

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 424
- Understanding How to Use Standard Firewall Filters on page 475

Stateless Firewall Filter Terms

- Stateless Firewall Filter Components on page 431
- Standard Firewall Filter Match Conditions for IPv4 Traffic on page 436
- Standard Firewall Filter Match Conditions for IPv6 Traffic on page 446
- Firewall Filter Match Conditions Based on Bit-Field Values on page 453
- Firewall Filter Match Conditions Based on Numbers or Text Aliases on page 458
- Firewall Filter Match Conditions Based on Address Fields on page 459
- Firewall Filter Match Conditions Based on Address Classes on page 467
- Standard Firewall Filter Terminating Actions on page 468
- Standard Firewall Filter Nonterminating Actions on page 469

Stateless Firewall Filter Components

This topic covers the following information:

- Protocol Family on page 431
- Filter Type on page 432
- Terms on page 433
- Match Conditions on page 434
- Actions on page 435

Protocol Family

Under the **firewall** statement, you can specify the protocol family for which you want to filter traffic.

Table 16 on page 432 describes the firewall filter protocol families.

Table 16: Firewall Filter Protocol Families

Type of Traffic to Be Filtered	Protocol Family Configuration Statement	Comments
Protocol Independent	**family any**	All protocol families configured on a logical interface.
Internet Protocol version 4 (IPv4)	**family inet**	The **family inet** statement is optional for IPv4.
Internet Protocol version 6 (IPv6)	**family inet6**	
MPLS	**family mpls**	
MPLS-tagged IPv4	**family mpls**	Supports matching on IP addresses and ports, up to five MPLS stacked labels.
Virtual private LAN service (VPLS)	**family vpls**	
Layer 2 Circuit Cross-Connection	**family ccc**	
Layer 2 Bridging	**family bridge**	MX Series routers only.

Filter Type

Under the **family** ***family-name*** statement, you can specify the type and name of the filter you want to configure.

Table 17 on page 432 describes the firewall filter types.

Table 17: Filter Types

Filter Type	Filter Configuration Statement	Description
Stateless Firewall Filter	**filter** ***filter-name***	Filters the following traffic types: • Protocol independent • IPv4 • IPv6 • MPLS • MPLS-tagged IPv4 • VPLS • Layer 2 CCC • Layer 2 bridging (MX Series routers only)

Copyright © 2012, Juniper Networks, Inc.

Table 17: Filter Types *(continued)*

Filter Type	Filter Configuration Statement	Description
Service Filter	**service-filter** ***service-filter-name***	Defines packet-filtering to be applied to ingress or egress before it is accepted for service processing or applied to returning service traffic after service processing has completed. Filters the following traffic types: • IPv4 • IPv6 Supported at logical interfaces configured on the following hardware only: • Adaptive Services (AS) PICs on M Series and T Series routers • Multiservices (MS) PICs on M Series and T Series routers • Multiservices (MS) DPCs on MX Series routers
Simple Filter	**simple-filter** ***simple-filter-name***	Defines packet filtering to be applied to ingress traffic only. Filters the following traffic type: • IPv4 Supported at logical interfaces configured on the following hardware only: • Gigabit Ethernet Intelligent Queuing (IQ2) PICs installed on M120, M320, or T Series routers • Enhanced Queuing Dense Port Concentrators (EQ DPCs) installed on MX Series routers

Terms

Under the **filter**, **service-filter**, or **simple-filter** statement, you must configure at least one firewall filter *term*. A term is a named structure in which match conditions and actions are defined. Within a firewall filter, you must configure a unique name for each term.

TIP: You cannot apply a different filter on each direction of traffic on the same interface. As a result, it is common to create firewall filters with multiple terms.

All stateless firewall filters contain one or more terms, and each term consists of two components—match conditions and actions. The match conditions define the values or fields that the packet must contain to be considered a match. If a packet is a match, the corresponding action is taken. By default, a packet that does not match a firewall filter is discarded.

Copyright © 2012, Juniper Networks, Inc.

If a packet arrives on an interface for which no firewall filter is applied for the incoming traffic on that interface, the packet is accepted by default.

NOTE: **A firewall filter with a large number of terms can adversely affect both the configuration commit time and the performance of the Routing Engine.**

Additionally, you can configure a stateless firewall filter within the term of another filter. This method enables you to add common terms to multiple filters without having to modify all filter definitions. You can configure one filter with the desired common terms, and configure this filter as a term in other filters. Consequently, to make a change in these common terms, you need to modify only one filter that contains the common terms, instead of multiple filters.

Match Conditions

A firewall filter term must contain at least one packet-filtering criteria, called a *match condition*, to specify the field or value that a packet must contain in order to be considered a match for the firewall filter term. For a match to occur, the packet must match all the conditions in the term. If a packet matches a firewall filter term, the router takes the configured action on the packet.

If a firewall filter term contains multiple match conditions, a packet must meet *all* match conditions to be considered a match for the firewall filter term.

If a single match condition is configured with multiple values, such as a range of values, a packet must match only *one* of the values to be considered a match for the firewall filter term.

The scope of match conditions you can specify in a firewall filter term depends on the protocol family under which the firewall filter is configured. You can define various match conditions, including the IP source address field, IP destination address field, TCP or UDP source port field, IP protocol field, Internet Control Message Protocol (ICMP) packet type, IP options, TCP flags, incoming logical or physical interface, and outgoing logical or physical interface.

Each protocol family supports a different set of match conditions, and some match conditions are supported only on certain routing devices. For example, a number of match conditions for VPLS traffic are supported only on the MX Series 3D Universal Edge Routers.

In the **from** statement in a firewall filter term, you specify characteristics that the packet must have for the action in the subsequent **then** statement to be performed. The characteristics are referred to as *match conditions*. The packet must match all conditions in the **from** statement for the action to be performed, which also means that the order of the conditions in the **from** statement is not important.

If an individual match condition can specify a list of values (such as multiple source and destination addresses) or a range of numeric values, a match occurs if any of the values matches the packet.

 Copyright © 2012, Juniper Networks, Inc.

If a filter term does not specify match conditions, the term accepts all packets and the actions specified in the term's **then** statement are optional.

NOTE:
Some of the numeric range and bit-field match conditions allow you to specify a text synonym. For a complete list of synonyms:

- If you are using the J-Web interface, select the synonym from the appropriate list.
- If you are using the CLI, type a question mark (?) after the **from** statement.

Actions

The actions specified in a firewall filter term define the actions to take for any packet that matches the conditions specified in the term.

Actions that are configured within a single term are all taken on traffic that matches the conditions configured.

BEST PRACTICE: We strongly recommend that you explicitly configure one or more actions per firewall filter term. Any packet that matches all the conditions of the term is automatically accepted unless the term specifies other or additional actions.

Firewall filter actions fall into the following categories:

Filter-Terminating Actions

A filter-terminating action halts all evaluation of a firewall filter for a specific packet. The router performs the specified action, and no additional terms are examined.

Nonterminating Actions

Nonterminating actions are used to perform other functions on a packet, such as incrementing a counter, logging information about the packet header, sampling the packet data, or sending information to a remote host using the system log functionality.

The presence of a nonterminating action, such as **count**, **log**, or **syslog**, without an explicit terminating action, such as **accept**, **discard**, or **reject**, results in a default terminating action of **accept**. If you do not want the firewall filter action to terminate, use the **next term** action after the nonterminating action.

In this example, term 2 is never evaluated, because term 1 has the implicit default **accept** terminating action.

```
[edit firewall filter test]
term 1 {
   from {
      source-address {
         0.0.0.0/0;
```

```
        }
      }
      then {
        log;
        <accept> #By default if not specified
      }
    }
    term 2 {
      then {
        reject;
      }
    }
```

In this example, term 2 is evaluated, because term 1 has the explicit **next term** flow control action.

```
[edit firewall filter test]
term 1 {
  from {
    source-address {
      0.0.0.0/0;
    }
  }
  then {
    log;
    next term;
  }
}
term 2 {
  then {
    reject;
  }
}
```

Flow Control Action

For standard stateless firewall filters only, the action **next term** enables the router to perform configured actions on the packet and then evaluate the following term in the filter, rather than terminating the filter.

A maximum of 1024 **next term** actions are supported per standard stateless firewall filter configuration. If you configure a standard filter that exceeds this limit, your candidate configuration results in a commit error.

Related Documentation

- Stateless Firewall Filter Types on page 423
- "Inserting a New Identifier in a Junos Configuration" in the *Junos OS CLI User Guide*

Standard Firewall Filter Match Conditions for IPv4 Traffic

You can configure a standard stateless firewall filter with match conditions for Internet Protocol version 4 (IPv4) traffic (**family inet**). Table 18 on page 437 describes the ***match-conditions*** you can configure at the **[edit firewall family inet filter *filter-name* term *term-name* from]** hierarchy level.

 Copyright © 2012, Juniper Networks, Inc.

Table 18: Standard Firewall Filter Match Conditions for IPv4 Traffic

Match Condition	Description
address *address*	Match the IPv4 source or destination address field.
address *address* **except**	Do not match the IPv4 source or destination address field. NOTE: This match condition is not supported on PTX series packet transport switches.
ah-spi *spi-value*	(M Series routers, except M120 and M320) Match the IPsec authentication header (AH) security parameter index (SPI) value. NOTE: This match condition is not supported on PTX series packet transport switches.
ah-spi-except *spi-value*	(M Series routers, except M120 and M320) Do not match the IPsec AH SPI value. NOTE: This match condition is not supported on PTX series packet transport switches.
apply-groups	Specify which groups to inherit configuration data from. You can specify more than one group name. You must list them in order of inheritance priority. The configuration data in the first group takes priority over the data in subsequent groups.
apply-groups-except	Specify which groups not to inherit configuration data from. You can specify more than one group name.
destination-address *address*	Match the IPv4 destination address field. You cannot specify both the **address** and **destination-address** match conditions in the same term.
destination-address *address* **except**	Do not match the IPv4 destination address field. For more information, see the **destination-address** field. NOTE: This match condition is not supported on PTX series packet transport switches.
destination-class *class-names*	Match one or more specified destination class names (sets of destination prefixes grouped together and given a class name). For more information, see "Firewall Filter Match Conditions Based on Address Classes" on page 467. NOTE: This match condition is not supported on PTX series packet transport switches.
destination-class-except *class-names*	Do not match one or more specified destination class names. For details, see the **destination-class** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.

Copyright © 2012, Juniper Networks, Inc.

Table 18: Standard Firewall Filter Match Conditions for IPv4 Traffic *(continued)*

Match Condition	Description
destination-port *number*	Match the UDP or TCP destination port field. You cannot specify both the **port** and **destination-port** match conditions in the same term. If you configure this match condition, we recommend that you also configure the **protocol udp** or **protocol tcp** match statement in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): **afs** (1483), **bgp** (179), **biff** (512), **bootpc** (68), **bootps** (67), **cmd** (514), **cvspserver** (2401), **dhcp** (67), **domain** (53), **eklogin** (2105), **ekshell** (2106), **exec** (512), **finger** (79), **ftp** (21), **ftp-data** (20), **http** (80), **https** (443), **ident** (113), **imap** (143), **kerberos-sec** (88), **klogin** (543), **kpasswd** (761), **krb-prop** (754), **krbupdate** (760), **kshell** (544), **ldap** (389), **ldp** (646), **login** (513), **mobileip-agent** (434), **mobilip-mn** (435), **msdp** (639), **netbios-dgm** (138), **netbios-ns** (137), **netbios-ssn** (139), **nfsd** (2049), **nntp** (119), **ntalk** (518), **ntp** (123), **pop3** (110), **pptp** (1723), **printer** (515), **radacct** (1813), **radius** (1812), **rip** (520), **rkinit** (2108), **smtp** (25), **snmp** (161), **snmptrap** (162), **snpp** (444), **socks** (1080), **ssh** (22), **sunrpc** (111), **syslog** (514), **tacacs** (49), **tacacs-ds** (65), **talk** (517), **telnet** (23), **tftp** (69), **timed** (525), **who** (513), or **xdmcp** (177).
destination-port-except *number*	Do not match the UDP or TCP destination port field. For details, see the **destination-port** match condition.
destination-prefix-list *name*	Match destination prefixes in the specified list. Specify the name of a prefix list defined at the [**edit policy-options prefix-list** *prefix-list-name*] hierarchy level.
destination-prefix-list *name* **except**	Do not match destination prefixes in the specified list. For more information, see the **destination-prefix-list** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.

dscp *number*	Match the Differentiated Services code point (DSCP). The DiffServ protocol uses the type-of-service (ToS) byte in the IP header. The most significant 6 bits of this byte form the DSCP. For more information, see the *Junos OS Class of Service Configuration Guide*.
	
	You can specify a numeric value from **0** through **63**. To specify the value in hexadecimal form, include **0x** as a prefix. To specify the value in binary form, include **b** as a prefix.
	
	In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed):
	
	• RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*, defines one code point: **ef** (46).
	• RFC 2597, *Assured Forwarding PHB Group*, defines 4 classes, with 3 drop precedences in each class, for a total of 12 code points:
	  ◦ **af11** (10), **af12** (12), **af13** (14)
	  ◦ **af21** (18), **af22** (20), **af23** (22)
	  ◦ **af31** (26), **af32** (28), **af33** (30)
	  ◦ **af41** (34), **af42** (36), **af43** (38)

dscp-except *number*	Do not match on the DSCP number. For more information, see the **dscp** match condition.

Copyright © 2012, Juniper Networks, Inc.

Table 18: Standard Firewall Filter Match Conditions for IPv4 Traffic *(continued)*

Match Condition	Description
esp-spi ***spi-value***	Match the IPsec encapsulating security payload (ESP) SPI value. Match on this specific SPI value. You can specify the ESP SPI value in hexadecimal, binary, or decimal form. NOTE: This match condition is not supported on PTX series packet transport switches.
esp-spi-except ***spi-value***	Match the IPsec ESP SPI value. Do not match on this specific SPI value. NOTE: This match condition is not supported on PTX series packet transport switches.
first-fragment	Match if the packet is the first fragment of a fragmented packet. Do not match if the packet is a trailing fragment of a fragmented packet. The first fragment of a fragmented packet has a fragment offset value of **0**. This match condition is an alias for the bit-field match condition **fragment-offset 0** match condition. To match both first and trailing fragments, you can use two terms that specify different match conditions: **first-fragment** and **is-fragment**. NOTE: This match condition is not supported on PTX series packet transport switches.
forwarding-class ***class***	Match the forwarding class of the packet. Specify **assured-forwarding**, **best-effort**, **expedited-forwarding**, or **network-control**. For information about forwarding classes and router-internal output queues, see the *Junos OS Class of Service Configuration Guide*.
forwarding-class-except ***class***	Do not match the forwarding class of the packet. For details, see the **forwarding-class** match condition.
fragment-flags ***number***	(Ingress only) Match the three-bit IP fragmentation flags field in the IP header. In place of the numeric field value, you can specify one of the following keywords (the field values are also listed): **dont-fragment** (0x4), **more-fragments** (0x2), or **reserved** (0x8).
fragment-offset ***value***	Match the 13-bit fragment offset field in the IP header. The value is the offset, in 8-byte units, in the overall datagram message to the data fragment. Specify a numeric value, a range of values, or a set of values. An offset value of **0** indicates the first fragment of a fragmented packet. The **first-fragment** match condition is an alias for the **fragment-offset 0** match condition. To match both first and trailing fragments, you can use two terms that specify different match conditions (**first-fragment** and **is-fragment**).
fragment-offset-except ***number***	Do not match the 13-bit fragment offset field.

Copyright © 2012, Juniper Networks, Inc.

Table 18: Standard Firewall Filter Match Conditions for IPv4 Traffic *(continued)*

Match Condition	Description
icmp-code *number*	Match the ICMP message code field. If you configure this match condition, we recommend that you also configure the **protocol icmp** match condition in the same term. If you configure this match condition, you must also configure the **icmp-type** ***message-type*** match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated: • parameter-problem: **ip-header-bad** (0), **required-option-missing** (1) • redirect: **redirect-for-host** (1), **redirect-for-network** (0), **redirect-for-tos-and-host** (3), **redirect-for-tos-and-net** (2) • time-exceeded: **ttl-eq-zero-during-reassembly** (1), **ttl-eq-zero-during-transit** (0) • unreachable: **communication-prohibited-by-filtering** (13), **destination-host-prohibited** (10), **destination-host-unknown** (7), **destination-network-prohibited** (9), **destination-network-unknown** (6), **fragmentation-needed** (4), **host-precedence-violation** (14), **host-unreachable** (1), **host-unreachable-for-TOS** (12), **network-unreachable** (0), **network-unreachable-for-TOS** (11), **port-unreachable** (3), **precedence-cutoff-in-effect** (15), **protocol-unreachable** (2), **source-host-isolated** (8), **source-route-failed** (5)
icmp-code-except ***message-code***	Do not match the ICMP message code field. For details, see the **icmp-code** match condition.
icmp-type *number*	Match the ICMP message type field. If you configure this match condition, we recommend that you also configure the **protocol icmp** match condition in the same term. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): **echo-reply** (0), **echo-request** (8), **info-reply** (16), **info-request** (15), **mask-request** (17), **mask-reply** (18), **parameter-problem** (12), **redirect** (5), **router-advertisement** (9), **router-solicit** (10), **source-quench** (4), **time-exceeded** (11), **timestamp** (13), **timestamp-reply** (14), or **unreachable** (3).
icmp-type-except ***message-type***	Do not match the ICMP message type field. For details, see the **icmp-type** match condition.
interface ***interface-name***	Match the interface on which the packet was received. NOTE: If you configure this match condition with an interface that does not exist, the term does not match any packet.

Copyright © 2012, Juniper Networks, Inc.

Table 18: Standard Firewall Filter Match Conditions for IPv4 Traffic *(continued)*

Match Condition	Description
interface-group *group-number*	Match the logical interface on which the packet was received to the specified interface group or set of interface groups. For ***group-number***, specify a single value or a range of values from **0** through **255**. To assign a logical interface to an interface group ***group-number***, specify the ***group-number*** at the **[interfaces** ***interface-name*** **unit** ***number*** **family** ***family*** **filter group]** hierarchy level. NOTE: This match condition is not supported on PTX series packet transport switches. For more information, see Filtering Packets Received on a Set of Interface Groups Overview.
interface-group-except *group-number*	Do not match the logical interface on which the packet was received to the specified interface group or set of interface groups. For details, see the **interface-group** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
interface-set *interface-set-name*	Match the interface on which the packet was received to the specified interface set. To define an interface set, include the **interface-set** statement at the **[edit firewall]** hierarchy level. NOTE: This match condition is not supported on PTX series packet transport switches. For more information, see Filtering Packets Received on an Interface Set Overview.

Table 18: Standard Firewall Filter Match Conditions for IPv4 Traffic *(continued)*

Match Condition	Description
ip-options ***values***	Match the 8-bit IP option field, if present, to the specified value or list of values. In place of a numeric value, you can specify one of the following text synonyms (the option values are also listed): **loose-source-route** (131), **record-route** (7), **router-alert** (148), **security** (130), **stream-id** (136),**strict-source-route** (137), or **timestamp** (68). To match *any* value for the IP option, use the text synonym **any**. To match on *multiple* values, specify the list of values within square brackets ('**[**' and '**]**'). To match a *range* of values, use the value specification **[** ***value1-value2*** **]**. For example, the match condition **ip-options [0-147]** matches on an IP options field that contains the **loose-source-route**, **record-route**, or **security** values, or any other value from 0 through 147. However, this match condition does not match on an IP options field that contains only the **router-alert** value (148). For most interfaces, a filter term that specifies an **ip-option** match on one or more *specific* IP option values (a value other than **any**) causes packets to be sent to the Routing Engine so that the kernel can parse the IP option field in the packet header. • For a firewall filter term that specifies an **ip-option** match on one or more specific IP option values, you cannot specify the **count**, **log**, or **syslog** nonterminating actions *unless* you also specify the **discard** terminating action in the same term. This behavior prevents double-counting of packets for a filter applied to a transit interface on the router. • Packets processed on the kernel might be dropped in case of a system bottleneck. To ensure that matched packets are instead sent to the Packet Forwarding Engine (where packet processing is implemented in hardware), use the **ip-options any** match condition. The 10-Gigabit Ethernet Modular Port Concentrator (MPC), 100-Gigabit Ethernet MPC, 60-Gigabit Ethernet MPC, 60-Gigabit Queuing Ethernet MPC, and 60-Gigabit Ethernet Enhanced Queuing MPC on MX Series routers are capable of parsing the IP option field of the IPv4 packet header. For interfaces configured on those MPCs, *all* packets that are matched using the **ip-options** match condition are sent to the Packet Forwarding Engine for processing.
ip-options-except ***values***	Do not match the IP option field to the specified value or list of values. For details about specifying the ***values***, see the **ip-options** match condition.
is-fragment	Match if the packet is a trailing fragment of a fragmented packet. Do not match the first fragment of a fragmented packet. NOTE: To match both first and trailing fragments, you can use two terms that specify different match conditions (**first-fragment** and **is-fragment**).

Copyright © 2012, Juniper Networks, Inc.

Table 18: Standard Firewall Filter Match Conditions for IPv4 Traffic *(continued)*

Match Condition	Description
loss-priority ***level***	Match the packet loss priority (PLP) level. Specify a single level or multiple levels: **low**, **medium-low**, **medium-high**, or **high**. Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers. For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the **tri-color** statement at the **[edit class-of-service]** hierarchy level to commit a PLP configuration with any of the four levels specified. If the **tri-color** statement is not enabled, you can only configure the **high** and **low** levels. This applies to all protocol families. NOTE: This match condition is not supported on PTX series packet transport switches. For information about the **tri-color** statement and for information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the *Junos OS Class of Service Configuration Guide*.
loss-priority-except ***level***	Do not match the PLP level. For details, see the **loss-priority** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
packet-length ***bytes***	Match the length of the received packet, in bytes. The length refers only to the IP packet, including the packet header, and does not include any Layer 2 encapsulation overhead.
packet-length-except ***bytes***	Do not match the length of the received packet, in bytes. For details, see the **packet-length** match type.
port ***number***	Match the UDP or TCP source or destination port field. If you configure this match condition, you cannot configure the **destination-port** match condition or the **source-port** match condition in the same term. If you configure this match condition, we recommend that you also configure the **protocol udp** or **protocol tcp** match statement in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the text synonyms listed under **destination-port**.
port-except ***number***	Do not match the UDP or TCP source or destination port field. For details, see the **port** match condition.
precedence ***ip-precedence-field***	Match the IP precedence field. In place of the numeric field value, you can specify one of the following text synonyms (the field values are also listed): **critical-ecp** (0xa0), **flash** (0x60), **flash-override** (0x80), **immediate** (0x40), **internet-control** (0xc0), **net-control** (0xe0), **priority** (0x20), or **routine** (0x00). You can specify precedence in hexadecimal, binary, or decimal form.

Table 18: Standard Firewall Filter Match Conditions for IPv4 Traffic *(continued)*

Match Condition	Description
precedence-exceptip-precedence-field	Do notM match the IP precedence field. In place of the numeric field value, you can specify one of the following text synonyms (the field values are also listed): **critical-ecp** (0xa0), **flash** (0x60), **flash-override** (0x80), **immediate** (0x40), **internet-control** (0xc0), **net-control** (0xe0), **priority** (0x20), or **routine** (0x00). You can specify precedence in hexadecimal, binary, or decimal form.
prefix-list *name*	Match the prefixes of the source or destination address fields to the prefixes in the specified list. The prefix list is defined at the **[edit policy-options prefix-list** *prefix-list-name***]** hierarchy level.
prefix-list *name* **except**	Do not match the prefixes of the source or destination address fields to the prefixes in the specified list. For more information, see the **prefix-list** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
protocol *number*	Match the IP protocol type field. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): **ah** (51), **dstopts** (60), **egp** (8), **esp** (50), **fragment** (44), **gre** (47), **hop-by-hop** (0), **icmp** (1), **icmp6** (58), **icmpv6** (58), **igmp** (2), **ipip** (4), **ipv6** (41), **ospf** (89), **pim** (103), **rsvp** (46), **sctp** (132), **tcp** (6), **udp** (17), or **vrrp** (112).
protocol-except *number*	Do not match the IP protocol type field. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): **ah** (51), **dstopts** (60), **egp** (8), **esp** (50), **fragment** (44), **gre** (47), **hop-by-hop** (0), **icmp** (1), **icmp6** (58), **icmpv6** (58), **igmp** (2), **ipip** (4), **ipv6** (41), **ospf** (89), **pim** (103), **rsvp** (46), **sctp** (132), **tcp** (6), **udp** (17), or **vrrp** (112).
service-filter-hit	Match a packet received from a filter where a **service-filter-hit** action was applied. NOTE: This match condition is not supported on PTX series packet transport switches.
source-address *address*	Match the IPv4 address of the source node sending the packet. You cannot specify both the **address** and **source-address** match conditions in the same term.
source-address *address* **except**	Do not match the IPv4 address of the source node sending the packet. For more information, see the **source-address** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
source-class *class-names*	Match one or more specified source class names (sets of source prefixes grouped together and given a class name). For more information, see "Firewall Filter Match Conditions Based on Address Classes" on page 467. NOTE: This match condition is not supported on PTX series packet transport switches.
source-class-except *class-names*	Do not match one or more specified source class names. For details, see the **source-class** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.

Copyright © 2012, Juniper Networks, Inc.

Table 18: Standard Firewall Filter Match Conditions for IPv4 Traffic *(continued)*

Match Condition	Description
source-port *number*	Match the UDP or TCP source port field. You cannot specify the **port** and **source-port** match conditions in the same term. If you configure this match condition for IPv4 traffic, we recommend that you also configure the **protocol udp** or **protocol tcp** match statement in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the text synonyms listed with the **destination-port** *number* match condition.
source-port-except *number*	Do not match the UDP or TCP source port field. For details, see the **source-port** match condition.
source-prefix-list *name*	Match source prefixes in the specified list. Specify the name of a prefix list defined at the **[edit policy-options prefix-list** *prefix-list-name*] hierarchy level.
source-prefix-list *name* **except**	Do not match source prefixes in the specified list. For more information, see the **source-prefix-list** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
tcp-established	Match TCP packets of an established TCP session (packets other than the first packet of a connection). This is an alias for **tcp-flags "(ack | rst)"**. This match condition does not implicitly check that the protocol is TCP. To check this, specify the **protocol tcp** match condition.
tcp-flags *value*	Match one or more of the low-order 6 bits in the 8-bit TCP flags field in the TCP header. To specify individual bit fields, you can specify the following text synonyms or hexadecimal values: • **fin** (0x01) • **syn** (0x02) • **rst** (0x04) • **push** (0x08) • **ack** (0x10) • **urgent** (0x20) In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet. You can string together multiple flags using the bit-field logical operators. For combined bit-field match conditions, see the **tcp-established** and **tcp-initial** match conditions. If you configure this match condition, we recommend that you also configure the **protocol tcp** match statement in the same term to specify that the TCP protocol is being used on the port. For IPv4 traffic only, this match condition does not implicitly check whether the datagram contains the first fragment of a fragmented packet. To check for this condition for IPv4 traffic only, use the **first-fragment** match condition.

Copyright © 2012, Juniper Networks, Inc.

Table 18: Standard Firewall Filter Match Conditions for IPv4 Traffic *(continued)*

Match Condition	Description
tcp-initial	Match the initial packet of a TCP connection. This is an alias for **tcp-flags "(!ack & syn)"**. This condition does not implicitly check that the protocol is TCP. If you configure this match condition, we recommend that you also configure the **protocol tcp** match condition in the same term.
ttl *number*	Match the IPv4 time-to-live number. Specify a TTL value or a range of TTL values. For *number*, you can specify one or more values from **0** through **255**. This match condition is supported only on M120, M320, MX Series, and T Series routers.
ttl-except *number*	Do not match on the IPv4 TTL number. For details, see the **ttl** match condition.
vlan-ether-type *value*	Match the virtual local area network (VLAN) Ethernet type field of a VPLS packet. NOTE: This match condition is not supported on PTX series packet transport switches.
vlan-ether-type-except *value*	Do not match the VLAN Ethernet type field of a VPLS packet. NOTE: This match condition is not supported on PTX series packet transport switches.

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 424
- Standard Firewall Filter Terminating Actions on page 468
- Standard Firewall Filter Nonterminating Actions on page 469

Standard Firewall Filter Match Conditions for IPv6 Traffic

You can configure a standard stateless firewall filter with match conditions for Internet Protocol version 6 (IPv6) traffic (**family inet6**). Table 19 on page 446 describes the ***match-conditions*** you can configure at the **[edit firewall family inet6 filter *filter-name* term *term-name* from]** hierarchy level.

Table 19: Standard Firewall Filter Match Conditions for IPv6 Traffic

Match Condition	Description
address *address*	Match the IPv6 source or destination address field.
address *address* **except**	Do not match the IPv6 source or destination address field. NOTE: This match condition is not supported on PTX series packet transport switches.
apply-groups	Specify which groups to inherit configuration data from. You can specify more than one group name. You must list them in order of inheritance priority. The configuration data in the first group takes priority over the data in subsequent groups.
apply-groups-except	Specify which groups not to inherit configuration data from. You can specify more than one group name.

Copyright © 2012, Juniper Networks, Inc.

Table 19: Standard Firewall Filter Match Conditions for IPv6 Traffic *(continued)*

Match Condition	Description
destination-address *address*	Match the IPv6 destination address field. You cannot specify both the **address** and **destination-address** match conditions in the same term.
destination-address *address* **except**	Do not match the IPv6 destination address field. For more information, see the **destination-address** field. NOTE: This match condition is not supported on PTX series packet transport switches.
destination-class *class-names*	Match one or more specified destination class names (sets of destination prefixes grouped together and given a class name). NOTE: This match condition is not supported on PTX series packet transport switches. For more information, see "Firewall Filter Match Conditions Based on Address Classes" on page 467.
destination-class-except *class-names*	Do not match one or more specified destination class names. For details, see the **destination-class** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
destination-port *number*	Match the UDP or TCP destination port field. You cannot specify both the **port** and **destination-port** match conditions in the same term. If you configure this match condition, we recommend that you also configure the **next-header udp** or **next-header tcp** match condition in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): **afs** (1483), **bgp** (179), **biff** (512), **bootpc** (68), **bootps** (67), **cmd** (514), **cvspserver** (2401), **dhcp** (67), **domain** (53), **eklogin** (2105), **ekshell** (2106), **exec** (512), **finger** (79), **ftp** (21), **ftp-data** (20), **http** (80), **https** (443), **ident** (113), **imap** (143), **kerberos-sec** (88), **klogin** (543), **kpasswd** (761), **krb-prop** (754), **krbupdate** (760), **kshell** (544), **ldap** (389), **ldp** (646), **login** (513), **mobileip-agent** (434), **mobilip-mn** (435), **msdp** (639), **netbios-dgm** (138), **netbios-ns** (137), **netbios-ssn** (139), **nfsd** (2049), **nntp** (119), **ntalk** (518), **ntp** (123), **pop3** (110), **pptp** (1723), **printer** (515), **radacct** (1813), **radius** (1812), **rip** (520), **rkinit** (2108), **smtp** (25), **snmp** (161), **snmptrap** (162), **snpp** (444), **socks** (1080), **ssh** (22), **sunrpc** (111), **syslog** (514), **tacacs** (49), **tacacs-ds** (65), **talk** (517), **telnet** (23), **tftp** (69), **timed** (525), **who** (513), or **xdmcp** (177).
destination-port-except *number*	Do not match the UDP or TCP destination port field. For details, see the **destination-port** match condition.
destination-prefix-list *prefix-list-name*	Match the prefix of the IPv6 destination address field. The prefix list is defined at the **[edit policy-options prefix-list** *prefix-list-name*] hierarchy level.
destination-prefix-list *prefix-list-name* **except**	Do not match the prefix of the IPv6 destination address field. For more information, see the **destination-prefix-list** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.

Table 19: Standard Firewall Filter Match Conditions for IPv6 Traffic *(continued)*

Match Condition	Description
forwarding-class *class*	Match the forwarding class of the packet. Specify **assured-forwarding**, **best-effort**, **expedited-forwarding**, or **network-control**. For information about forwarding classes and router-internal output queues, see the *Junos OS Class of Service Configuration Guide*.
forwarding-class-except *class*	Do not match the forwarding class of the packet. For details, see the **forwarding-class** match condition.
icmp-code *message-code*	Match the ICMP message code field. If you configure this match condition, we recommend that you also configure the **next-header icmp** or **next-header icmp6** match condition in the same term. If you configure this match condition, you must also configure the **icmp-type** *message-type* match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated: • parameter-problem: **ip6-header-bad** (0), **unrecognized-next-header** (1), **unrecognized-option** (2) • time-exceeded: **ttl-eq-zero-during-reassembly** (1), **ttl-eq-zero-during-transit** (0) • destination-unreachable: **administratively-prohibited** (1), **address-unreachable** (3), **no-route-to-destination** (0), **port-unreachable** (4)
icmp-code-except *message-code*	Do not match the ICMP message code field. For details, see the **icmp-code** match condition.
icmp-type *message-type*	Match the ICMP message type field. If you configure this match condition, we recommend that you also configure the **next-header icmp** or **next-header icmp6** match condition in the same term. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): **destination-unreachable** (1), **echo-reply** (129), **echo-request** (128), **membership-query** (130), **membership-report** (131), **membership-termination** (132), **neighbor-advertisement** (136), **neighbor-solicit** (135), **node-information-reply** (140), **node-information-request** (139), **packet-too-big** (2), **parameter-problem** (4), **redirect** (137), **router-advertisement** (134), **router-renumbering** (138), **router-solicit** (133), or **time-exceeded** (3).
icmp-type-except *message-type*	Do not match the ICMP message type field. For details, see the **icmp-type** match condition.
interface *interface-name*	Match the interface on which the packet was received. NOTE: If you configure this match condition with an interface that does not exist, the term does not match any packet.

Copyright © 2012, Juniper Networks, Inc.

Table 19: Standard Firewall Filter Match Conditions for IPv6 Traffic *(continued)*

Match Condition	Description
interface-group *group-number*	Match the logical interface on which the packet was received to the specified interface group or set of interface groups. For ***group-number***, specify a single value or a range of values from **0** through **255**. To assign a logical interface to an interface group ***group-number***, specify the ***group-number*** at the **[interfaces** ***interface-name*** **unit** ***number*** **family** ***family*** **filter group]** hierarchy level. For more information, see Filtering Packets Received on a Set of Interface Groups Overview.
interface-group-except *group-number*	Do not match the logical interface on which the packet was received to the specified interface group or set of interface groups. For details, see the **interface-group** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
interface-set *interface-set-name*	Match the interface on which the packet was received to the specified interface set. To define an interface set, include the **interface-set** statement at the **[edit firewall]** hierarchy level. NOTE: This match condition is not supported on PTX series packet transport switches. For more information, see Filtering Packets Received on an Interface Set Overview.
loss-priority *level*	Match the packet loss priority (PLP) level. Specify a single level or multiple levels: **low**, **medium-low**, **medium-high**, or **high**. Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers. For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the **tri-color** statement at the **[edit class-of-service]** hierarchy level to commit a PLP configuration with any of the four levels specified. If the **tri-color** statement is not enabled, you can only configure the **high** and **low** levels. This applies to all protocol families. NOTE: This match condition is not supported on PTX series packet transport switches. For information about the **tri-color** statement and for information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the *Junos OS Class of Service Configuration Guide*.
loss-priority-except *level*	Do not match the PLP level. For details, see the **loss-priority** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.

Table 19: Standard Firewall Filter Match Conditions for IPv6 Traffic *(continued)*

Match Condition	Description
next-header ***header-type***	Match the 8-bit IP protocol field that identifies the type of header immediately following the IP header. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): **ah** (51), **dstops** (60), **egp** (8), **esp** (50), **fragment** (44), **gre** (47), **hop-by-hop** (0), **icmp** (1), **icmp6** (58), **icmpv6** (58), **igmp** (2), **ipip** (4), **ipv6** (41), **no-next-header** (59), **ospf** (89), **pim** (103), **routing** (43), **rsvp** (46), **sctp** (132), **tcp** (6), **udp** (17), or **vrrp** (112). NOTE: **next-header icmp6** and **next-header icmpv6** match conditions perform the same function. **next-header icmp6** is the preferred option. **next-header icmpv6** is hidden in the Junos OS CLI.
next-header-except ***header-type***	Do not match the 8-bit IP protocol field that identifies the type of header immediately following the IPv6 header. For details, see the **next-header** match type.
packet-length ***bytes***	Match the length of the received packet, in bytes. The length refers only to the IP packet, including the packet header, and does not include any Layer 2 encapsulation overhead.
packet-length-except ***bytes***	Do not match the length of the received packet, in bytes. For details, see the **packet-length** match type.
port ***number***	Match the UDP or TCP source or destination port field. If you configure this match condition, you cannot configure the **destination-port** match condition or the **source-port** match condition in the same term. If you configure this match condition, we recommend that you also configure the **next-header udp** or **next-header tcp** match condition in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the text synonyms listed under **destination-port**.
port-except ***number***	Do not match the UDP or TCP source or destination port field. For details, see the **port** match condition.
prefix-list ***prefix-list-name***	Match the prefixes of the source or destination address fields to the prefixes in the specified list. The prefix list is defined at the **[edit policy-options prefix-list** ***prefix-list-name*****]** hierarchy level.
prefix-list ***prefix-list-name*** **except**	Do not match the prefixes of the source or destination address fields to the prefixes in the specified list. For more information, see the **prefix-list** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
service-filter-hit	Match a packet received from a filter where a **service-filter-hit** action was applied. NOTE: This match condition is not supported on PTX series packet transport switches.
source-address ***address***	Match the IPv6 address of the source node sending the packet. You cannot specify both the **address** and **source-address** match conditions in the same term.

Copyright © 2012, Juniper Networks, Inc.

Table 19: Standard Firewall Filter Match Conditions for IPv6 Traffic *(continued)*

Match Condition	Description
source-address ***address*** **except**	Do not match the IPv6 address of the source node sending the packet. For more information, see the **source-address** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
source-class ***class-names***	Match one or more specified source class names (sets of source prefixes grouped together and given a class name). NOTE: This match condition is not supported on PTX series packet transport switches. For more information, see "Firewall Filter Match Conditions Based on Address Classes" on page 467.
source-class-except ***class-names***	Do not match one or more specified source class names. For details, see the **source-class** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
source-port ***number***	Match the UDP or TCP source port field. You cannot specify the **port** and **source-port** match conditions in the same term. If you configure this match condition, we recommend that you also configure the **next-header udp** or **next-header tcp** match condition in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the text synonyms listed with the **destination-port** ***number*** match condition.
source-port-except ***number***	Do not match the UDP or TCP source port field. For details, see the **source-port** match condition.
source-prefix-list ***name***	Match the IPv6 address prefix of the packet source field. Specify a prefix list name defined at the **[edit policy-options prefix-list** ***prefix-list-name*****]** hierarchy level.
source-prefix-list ***name*** **except**	Do not match the IPv6 address prefix of the packet source field. For more information, see the **source-prefix-list** match condition. NOTE: This match condition is not supported on PTX series packet transport switches.
tcp-established	Match TCP packets other than the first packet of a connection. This is a text synonym for **tcp-flags "(ack | rst)"** (**0x14**). NOTE: This condition does not implicitly check that the protocol is TCP. To check this, specify the **protocol tcp** match condition. If you configure this match condition, we recommend that you also configure the **next-header tcp** match condition in the same term.

Table 19: Standard Firewall Filter Match Conditions for IPv6 Traffic *(continued)*

Match Condition	Description
tcp-flags ***flags***	Match one or more of the low-order 6 bits in the 8-bit TCP flags field in the TCP header. To specify individual bit fields, you can specify the following text synonyms or hexadecimal values: • **fin** (0x01) • **syn** (0x02) • **rst** (0x04) • **push** (0x08) • **ack** (0x10) • **urgent** (0x20) In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet. You can string together multiple flags using the bit-field logical operators. For combined bit-field match conditions, see the **tcp-established** and **tcp-initial** match conditions. If you configure this match condition, we recommend that you also configure the **next-header tcp** match condition in the same term to specify that the TCP protocol is being used on the port.
tcp-initial	Match the initial packet of a TCP connection. This is a text synonym for **tcp-flags "(!ack & syn)"**. This condition does not implicitly check that the protocol is TCP. If you configure this match condition, we recommend that you also configure the **next-header tcp** match condition in the same term.

traffic-class ***number***	Match the 8-bit field that specifies the class-of-service (CoS) priority of the packet.
	
	This field was previously used as the type-of-service (ToS) field in IPv4.
	
	You can specify a numeric value from **0** through **63**. To specify the value in hexadecimal form, include **0x** as a prefix. To specify the value in binary form, include **b** as a prefix.
	
	In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed):
	
	• RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*, defines one code point: **ef** (46).
	• RFC 2597, *Assured Forwarding PHB Group*, defines 4 classes, with 3 drop precedences in each class, for a total of 12 code points:
	  ◦ **af11** (10), **af12** (12), **af13** (14)
	  ◦ **af21** (18), **af22** (20), **af23** (22)
	  ◦ **af31** (26), **af32** (28), **af33** (30)
	  ◦ **af41** (34), **af42** (36), **af43** (38)

traffic-class-except ***number***	Do not match the 8-bit field that specifies the CoS priority of the packet. For details, see the **traffic-class** match description.

Copyright © 2012, Juniper Networks, Inc.

NOTE: If you specify an IPv6 address in a match condition (the **address**, **destination-address**, or **source-address** match conditions), use the syntax for text representations described in RFC 2373, *IP Version 6 Addressing Architecture*. For more information about IPv6 addresses, see "IPv6 Overview" and "IPv6 Standards" in the *Junos OS Routing Protocols Configuration Guide*.

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 424
- Standard Firewall Filter Terminating Actions on page 468
- Standard Firewall Filter Nonterminating Actions on page 469

Firewall Filter Match Conditions Based on Bit-Field Values

- Match Conditions for Bit-Field Values on page 453
- Match Conditions for Common Bit-Field Values or Combinations on page 454
- Logical Operators for Bit-Field Values on page 455
- Matching on a Single Bit-Field Value or Text Alias on page 456
- Matching on Multiple Bit-Field Values or Text Aliases on page 456
- Matching on a Negated Bit-Field Value on page 456
- Matching on the Logical OR of Two Bit-Field Values on page 457
- Matching on the Logical AND of Two Bit-Field Values on page 457
- Grouping Bit-Field Match Conditions on page 457

Match Conditions for Bit-Field Values

Table 20 on page 453 lists the firewall filter match conditions that are based on whether certain bit fields in a packet are set or not set. The second and third columns list the types of traffic for which the match condition is supported.

Table 20: Binary and Bit-Field Match Conditions for Firewall Filters

Bit-Field Match Condition	Match Values	Protocol Families for Standard Stateless Firewall Filters	Protocol Families for Service Filters
fragment-flags ***flags***	Hexadecimal values or text aliases for the three-bit IP fragmentation flags field in the IP header.	**family inet**	**family inet**

† The Junos OS does not automatically check the first fragment bit when matching TCP flags for IPv4 traffic. To check the first fragment bit for IPv4 traffic only, use the **first-fragment** match condition.

Table 20: Binary and Bit-Field Match Conditions for Firewall Filters *(continued)*

Bit-Field Match Condition	Match Values	Protocol Families for Standard Stateless Firewall Filters	Protocol Families for Service Filters
fragment-offset *value*	Hexadecimal values or text aliases for the 13-bit fragment offset field in the IP header.	**family inet**	**family inet**
tcp-flags *value*†	Hexadecimal values or text aliases for the low-order 6 bits of the 8-bit TCP flags field in the TCP header.	**family inet** **family inet6** **family vpls** **family bridge**	**family inet** **family inet6**

† The Junos OS does not automatically check the first fragment bit when matching TCP flags for IPv4 traffic. To check the first fragment bit for IPv4 traffic only, use the **first-fragment** match condition.

Match Conditions for Common Bit-Field Values or Combinations

Table 21 on page 454 describes firewall filter match conditions that are based on whether certain commonly used values or *combinations* of bit fields in a packet are set or not set.

You can use text synonyms to specify some common bit-field matches. In the previous example, you can specify **tcp-initial** as the same match condition.

NOTE:

Some of the numeric range and bit-field match conditions allow you to specify a text synonym. For a complete list of synonyms:

- If you are using the J-Web interface, select the synonym from the appropriate list.
- If you are using the CLI, type a question mark (**?**) after the **from** statement.

Table 21: Bit-Field Match Conditions for Common Combinations

Match Condition	Description	Protocol Families for Standard Stateless Firewall Filters	Protocol Families for Service Filters
first-fragment	Text alias for the bit-field match condition **fragment-offset 0**, which indicates the first fragment of a fragmented packet.	**family inet**	**family inet**

Copyright © 2012, Juniper Networks, Inc.

Table 21: Bit-Field Match Conditions for Common Combinations *(continued)*

Match Condition	Description	Protocol Families for Standard Stateless Firewall Filters	Protocol Families for Service Filters
is-fragment	Text alias for the bit-field match condition **fragment-offset 0 except**, which indicates a trailing fragment of a fragmented packet.	**family inet**	**family inet**
tcp-established	Alias for the bit-field match condition **tcp-flags "(ack | rst)"**, which indicates an established TCP session, but not the first packet of a TCP connection.	**family inet** **family inet6**	—
tcp-initial	Alias for the bit-field match condition **tcp-flags "(!ack & syn)"**, which indicates the first packet of a TCP connection, but not an established TCP session.	**family inet** **family inet6**	—

Logical Operators for Bit-Field Values

Table 22 on page 455 lists the logical operators you can apply to *single* bit-field values when specifying stateless firewall filter match conditions. The operators are listed in order, from highest precedence to lowest precedence. Operations are left-associative, meaning that the operations are processed from left to right.

Table 22: Bit-Field Logical Operators

Precedence Order	Bit-Field Logical Operator	Description
1	**(*complex-match-condition*)**	Grouping—The complex match condition is evaluated before any operators outside the parentheses are applied.
2	**!** ***match-condition***	Negation—A match occurs if the match condition is false.
3	***match-condition-1* & *match-condition-2*** or ***match-condition-1* + *match-condition-2***	Logical AND—A match occurs if both match conditions are true.
4	***match-condition-1* | *match-condition-2*** or ***match-condition-1* , *match-condition-2***	Logical OR—A match occurs if either match condition is true.

Matching on a Single Bit-Field Value or Text Alias

For the **fragment-flags** and **tcp-flags** bit-match conditions, you can specify firewall filter match conditions based on whether a particular bit in the packet field is set or not set.

- Numeric value to specify a single bit—You can specify a single bit-field match condition by using a numeric value that has one bit set. Depending on the match condition, you can specify a decimal value, a binary value, or a hexadecimal value. To specify a binary value, specify the number with the prefix **b**. To specify a hexadecimal value, specify the number with the prefix **0x**.

  In the following example, a match occurs if the **RST** bit in the TCP flags field is set:

  ```
  [edit firewall family inet filter filter_tcp_rst_number term term1 from]
  user@host# set tcp-flags 0x04
  ```

- Text alias to specify a single bit—You generally specify a single bit-field match condition by using a text alias enclosed in double-quotation marks (“ ”).

  In the following example, a match occurs if the **RST** bit in the TCP flags field is set:

  ```
  [edit firewall family inet filter filter_tcp_rst_alias term term1 from]
  user@host# set tcp-flags “rst”
  ```

Matching on Multiple Bit-Field Values or Text Aliases

You can specify a firewall filter match condition based on whether a particular set of bits in a packet field are set.

- Numeric values to specify multiple set bits—When you specify a numeric value whose binary representation has more than one set bit, the value is treated as a logical AND of the set bits.

  In the following example, the two match conditions are the same. A match occurs if either bit **0x01** or **0x02** is not set:

  ```
  [edit firewall family inet filter reset_or_not_initial_packet term term5 from]
  user@host# set tcp-flags “!0x3”
  user@host# set tcp-flags “!(0x01 & 0x02)”
  ```

- Text aliases that specify common bit-field matches—You can use text aliases to specify some common bit-field matches. You specify these matches as a single keyword.

  In the following example, the **tcp-established** condition, which is an alias for **“(ack | rst)”**, specifies that a match occurs on TCP packets other than the first packet of a connection:

  ```
  [edit firewall family inet filter reset_or_not_initial_packet term term6 from]
  user@host# set tcp-established
  ```

Matching on a Negated Bit-Field Value

To negate a match, precede the value with an exclamation point.

In the following example, a match occurs if the **RST** bit in the TCP flags field is *not* set:

```
[edit firewall family inet filter filter_tcp_rst term term1 from]
```


Copyright © 2012, Juniper Networks, Inc.

```
user@host# set tcp-flags “!rst”
```

Matching on the Logical OR of Two Bit-Field Values

You can use the *logical OR operator* (**|** or **,**) to specify that a match occurs if a bit field matches either of two bit-field values specified.

In the following example, a match occurs if the packet is *not* the initial packet in a TCP session:

```
[edit firewall family inet filter not_initial_packet term term3 from]
user@host# set tcp-flags "!syn | ack"
```

In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet. In a packet that is not the initial packet in a TCP session, either the SYN flag is not set or the ACK flag is set.

Matching on the Logical AND of Two Bit-Field Values

You can use the *logical AND operator* (**&** or **+**) to specify that a match occurs if a bit field matches both of two bit-field values specified.

In the following example, a match occurs if the packet is the initial packet in a TCP session:

```
[edit firewall family inet filter initial_packet term term2 from]
user@host# set tcp-flags “syn & !ack”
```

In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet. In a packet that is an initial packet in a TCP session, the SYN flag is set and the ACK flag is not set.

Grouping Bit-Field Match Conditions

You can use the *logical grouping notation* to specify that the complex match condition inside the parentheses is evaluated before any operators outside the parentheses are applied.

In the following example, a match occurs if the packet is a TCP reset or if the packet is not the initial packet in the TCP session:

```
[edit firewall family inet filter reset_or_not_initial_packet term term4 from]
user@host# set tcp-flags “!(syn & !ack) | rst”
```

In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet. In a packet that is *not* the initial packet in a TCP session, the SYN flag is not set and the ACK field is set.

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 424
- Firewall Filter Match Conditions Based on Numbers or Text Aliases on page 458
- Firewall Filter Match Conditions Based on Address Fields on page 459
- Firewall Filter Match Conditions Based on Address Classes on page 467

Firewall Filter Match Conditions Based on Numbers or Text Aliases

This topic covers the following information:

- Matching on a Single Numeric Value on page 458
- Matching on a Range of Numeric Values on page 458
- Matching on a Text Alias for a Numeric Value on page 458
- Matching on a List of Numeric Values or Text Aliases on page 458

Matching on a Single Numeric Value

You can specify a firewall filter match condition based on whether a particular packet field value is a specified numeric value. In the following example, a match occurs if the packet source port number is **25**:

```
[edit firewall family inet filter filter1 term term1 from]
user@host# set source-port 25
```

Matching on a Range of Numeric Values

You can specify a firewall filter match condition based on whether a particular packet field value falls within a specified range of numeric values. In the following example, a match occurs for source ports values from **1024** through **65,535**, inclusive:

```
[edit firewall family inet filter filter2 term term1 from]
user@host# set source-port 1024-65535
```

Matching on a Text Alias for a Numeric Value

You can specify a firewall filter match condition based on whether a particular packet field value is a numeric value that you specify by using a text string as an *alias* for the numeric value. In the following example, a match occurs if the packet source port number is **25**. For the **source-port** and **destination-port** match conditions, the text alias**smtp** corresponds to the numeric value **25**.

```
[edit firewall family inet filter filter3 term term1 from]
user@host# set source-port smtp
```

Matching on a List of Numeric Values or Text Aliases

You can specify a firewall filter match condition based on whether a particular packet field value matches any one of multiple numeric values or text aliases that you specify within square brackets and delimited by spaces. In the following example, a match occurs if the packet source port number is any of the following values: **20** (which corresponds to the text aliases **ftp-data**), **25**, or any value from **1024** through **65535**.

```
[edit firewall family inet filter filter3 term term1 from]
user@host# set source-port [ smtp ftp-data 25 1024-65535 ]
```

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 424
- Firewall Filter Match Conditions Based on Bit-Field Values on page 453
- Firewall Filter Match Conditions Based on Address Fields on page 459
- Firewall Filter Match Conditions Based on Address Classes on page 467

Copyright © 2012, Juniper Networks, Inc.

Firewall Filter Match Conditions Based on Address Fields

You can configure firewall filter match conditions that evaluate packet address fields—IPv4 source and destination addresses, IPv6 source and destination addresses, or media access control (MAC) source and destination addresses—against specified addresses or prefix values.

- Implied Match on the '0/0 except' Address for Firewall Filter Match Conditions Based on Address Fields on page 459
- Matching an Address Field to a Subnet Mask or Prefix on page 459
- Matching an Address Field to an Excluded Value on page 460
- Matching Either IP Address Field to a Single Value on page 464
- Matching an Address Field to Noncontiguous Prefixes on page 464
- Matching an Address Field to a Prefix List on page 466

Implied Match on the '0/0 except' Address for Firewall Filter Match Conditions Based on Address Fields

Every firewall filter match condition based on a set of addresses or address prefixes is associated with an implicit match on the address **0.0.0.0/0 except** (for IPv4 or VPLS traffic) or **0:0:0:0:0:0:0:0/0 except** (for IPv6 traffic). As a result, any packet whose specified address field does not match any of the specified addresses or address prefixes fails to match the entire term.

Matching an Address Field to a Subnet Mask or Prefix

You can specify a single match condition to match a source address or destination address that falls within a specified address prefix.

IPv4 Subnet Mask Notation

For an IPv4 address, you can specify a subnet mask value rather than a prefix length. For example:

```
[edit firewall family inet filter filter_on_dst_addr term term3 from]
user@host# set address 10.0.0.10/255.0.0.255
```

Prefix Notation

To specify the address prefix, use the notation ***prefix/prefix-length***. In the following example, a match occurs if a destination address matches the prefix **10.0.0.0/8**:

```
[edit firewall family inet filter filter_on_dst_addr term term1 from]
user@host# set destination-address 10.0.0.0/8
```

Default Prefix Length for IPv4 Addresses

If you do not specify ***/prefix-length*** for an IPv4 address, the prefix length defaults to **/32**. The following example illustrates the default prefix value:

```
[edit firewall family inet filter filter_on_dst_addr term term2 from]
user@host# set destination-address 10
user@host# show
destination-address {
```

```
    10.0.0.0/32;
  }
```

Default Prefix Length for IPv6 Addresses

If you do not specify ***/prefix-length*** for an IPv6 address, the prefix length defaults to **/128**. The following example illustrates the default prefix value:

```
[edit firewall family inet6 filter filter_on_dst_addr term term1 from]
user@host# set destination-address ::10
user@host# show
destination-address {
  ::10/128;
}
```

Default Prefix Length for MAC Addresses

If you do not specify ***/prefix-length*** for a media access control (MAC) address of a VPLS, Layer 2 CCC, or Layer 2 bridging packet, the prefix length defaults to **/48**. The following example illustrates the default prefix value:

```
[edit firewall family vpls filter filter_on_dst_mac_addr term term1 from]
user@host# set destination-mac-address 01:00:0c:cc:cc:cd
user@host# show
destination-address {
  01:00:0c:cc:cc:cd/48;
}
```

Matching an Address Field to an Excluded Value

For the address-field match conditions, you can include the **except** keyword to specify that a match occurs for an address field that does not match the specified address or prefix.

Excluding IP Addresses in IPv4 or IPv6 Traffic

For the following IPv4 and IPv6 match conditions, you can include the **except** keyword to specify that a match occurs for an IP address field that does not match the specified IP address or prefix:

- **address** ***address*** **except**—A match occurs if either the source IP address or the destination IP address does not match the specified address or prefix.
- **source-address** ***address*** **except**—A match occurs if the source IP address does not match the specified address or prefix.
- **destination-address** ***address*** **except**—A match occurs if the destination IP address does not match the specified address or prefix.

In the following example, a match occurs for any IPv4 destination addresses that fall under the **192.168.10.0/8** prefix, except for addresses that fall under **192.168.0.0/16**. All other addresses implicitly do not match this condition.

```
[edit firewall family inet filter filter_on_dst_addr term term1 from]
user@host# set 192.168.0.0/16 except
user@host# set 192.168.10.0/8
user@host# show
destination-address {
```

 Copyright © 2012, Juniper Networks, Inc.

```
    192.168.0.0/16 except;
    192.168.10.0/8;
  }
```

In the following example, a match occurs for any IPv4 destination address that does not fall within the prefix **10.1.1.0/24**:

```
[edit firewall family inet filter filter_on_dst_addr term term24 from]
user@host# set destination-address 0.0.0.0/0
user@host# set destination-address 10.1.1.0/24 except
user@host# show
destination-address {
    0.0.0.0/0;
    10.1.1.0/24 except;
}
```

Excluding IP Addresses in VPLS or Layer 2 Bridging Traffic

For the following VPLS and Layer 2 bridging match conditions on MX Series routers only, you can include the **except** keyword to specify that a match occurs for an IP address field that does not match the specified IP address or prefix:

- **ip-address** ***address*** **except**—A match occurs if either the source IP address or the destination IP address does not match the specified address or prefix.
- **source-ip-address** ***address*** **except**—A match occurs if the source IP address does not match the specified address or prefix.
- **destination-ip-address** ***address*** **except**—A match occurs if the destination IP address does not match the specified address or prefix.

In the following example for filtering VPLS traffic on an MX Series router, a match occurs if the source IP address falls within the exception range of **55.0.1.0/255.0.255.0** and the destination IP address matches **55.0.0.0/8**:

```
[edit]
firewall {
   family vpls {
      filter fvpls {
         term 1 {
            from {
               ip-address {
                  55.0.0.0/8;
                  55.0.1.0/255.0.255.0 except;
               }
            }
            then {
               count from-55/8;
               discard;
            }
         }
      }
   }
}
```

Excluding MAC Addresses in VPLS or Layer 2 Bridging Traffic

For the following VPLS or Layer 2 bridging traffic match conditions, you can include the **except** keyword to specify that a match occurs for a MAC address field that does not match the specified MAC address or prefix:

- **source-mac-address** ***address*** **except**—A match occurs if the source MAC address does not match the specified address or prefix.
- **destination-mac-address** ***address*** **except**—A match occurs if either the destination MAC address does not match the specified address or prefix.

Excluding All Addresses Requires an Explicit Match on the '0/0' Address

If you specify a firewall filter match condition that consists of one or more address-*exception* match conditions (address match conditions that use the **except** keyword) but no *matchable* address match conditions, packets that do not match any of the configured prefixes fails the overall match operation. To configure a firewall filter term of address-exception match conditions to match any address that is not in the prefix list, include an explicit match of **0/0** so that the term contain a matchable address.

For the following example firewall filter for IPv4 traffic, the **from-trusted-addresses** term fails to discard matching traffic, and the **INTRUDERS-COUNT** counter is missing from the output of the **show firewall** operational mode command:

```
[edit]
user@host# show policy-options
prefix-list TRUSTED-ADDRESSES {
   10.2.1.0/24;
   192.168.122.0/24;
}
```

 Copyright © 2012, Juniper Networks, Inc.

```
[edit firewall family inet filter protect-RE]
user@host# show
term from-trusted-addresses {
   from {
      source-prefix-list {
         TRUSTED-ADDRESSES except;
      }
      protocol icmp;
   }
   then {
      count INTRUDERS-COUNT;
      discard;
   }
}
   term other-icmp {
      from {
         protocol icmp;
      }
      then {
         count VALID-COUNT;
         accept;
      }
   }
   term all {
      then accept;
   }
```

```
[edit]
user@host#  run show firewall
Filter: protect-RE
Counters:
Name                                                    Bytes               Packets
VALID-COUNT                                              2770                    70
Filter: __default_bpdu_filter__
```

To cause a filter term of address-exception match conditions to match any address that is not in the prefix list, include an explicit match of **0/0** in the set of match conditions:

```
[edit firewall family inet filter protect-RE]
user@host# show term from-trusted-addresses
from {
   source-prefix-list {
      0.0.0.0/0;
      TRUSTED-ADDRESSES except;
   }
   protocol icmp;
}
```

With the addition of the **0.0.0.0/0** source prefix address to the match condition, the **from-trusted-addresses** term discards matching traffic, and the INTRUDERS-COUNT counter displays in the output of the **show firewall** operational mode command:

```
[edit]
user@host#  run show firewall
Filter: protect-RE
Counters:
Name                                                    Bytes               Packets
VALID-COUNT                                              2770                    70
```

```
INTRUDERS-COUNT                                    420                    5
Filter: __default_bpdu_filter__
```

Matching Either IP Address Field to a Single Value

For IPv4 and IPv6 traffic and for VPLS and Layer 2 bridging traffic on MX Series routers only, you can use a single match condition to match a single address or prefix value to either the source or destination IP address field.

Matching Either IP Address Field in IPv4 or IPv6 Traffic

For IPv4 or IPv6 traffic, you can use a single match condition to specify the same address or prefix value as the match for either the source or destination IP address field. Instead of creating separate filter terms that specify the same address for the **source-address** and **destination-address** match conditions, you use only the **address** match condition. A match occurs if *either* the source IP address *or* the destination IP address matches the specified address or prefix.

If you use the **except** keyword with the **address** match condition, a match occurs if *both* the source IP address and the destination IP address match the specified value *before* the exception applies.

In a firewall filter term that specifies either the **source-address** or the **destination-address** match condition, you cannot also specify the **address** match condition.

Matching Either IP Address Field in VPLS or Layer 2 Bridging Traffic

For VPLS or Layer 2 bridging traffic on MX Series routers only, you can use a single match condition to specify the same address or prefix value as the match for either the source or destination IP address field. Instead of creating separate filter terms that specify the same address for the **source-ip-address** and **destination-ip-address** match conditions, you use only the **ip-address** match condition. A match occurs if *either* the source IP address *or* the destination IP address matches the specified address or prefix.

If you use the **except** keyword with the **ip-address** match condition, a match occurs if *both* the source IP address and the destination IP address match the specified value *before* the exception applies.

In a firewall filter term that specifies either the **source-ip-address** or the **destination-ip-address** match condition, you cannot also specify the **ip-address** match condition.

Matching an Address Field to Noncontiguous Prefixes

For IPv4 traffic only, specify a single match condition to match the IP source or destination address field to any prefix specified. The prefixes do not need to be contiguous. That is, the prefixes under the **source-address** or **destination-address** match condition do not need to be adjacent or neighboring to one another.

In the following example, a match occurs if a destination address matches either the **10.0.0.0/8** prefix or the **192.168.0.0/32** prefix:

```
[edit firewall family inet filter filter_on_dst_addr term term5 from]
user@host# set destination-address 10.0.0.0/8
```


Copyright © 2012, Juniper Networks, Inc.

```
user@host# set destination-address 192.168.0.0/32
user@host# show
destination-address {
   destination-address 10.0.0.0/8;
   destination-address 192.168.0.0/32;
}
```

The order in which you specify the prefixes within the match condition is not significant. Packets are evaluated against all the prefixes in the match condition to determine whether a match occurs. If prefixes overlap, longest-match rules are used to determine whether a match occurs. A match condition of noncontiguous prefixes includes an implicit **0/0 except** statement, which means that any prefix that does not match any prefix included in the match condition is explicitly considered not to match.

Because the prefixes are order-independent and use longest-match rules, longer prefixes subsume shorter ones as long as they are the same type (whether you specify **except** or not). This is because anything that would match the longer prefix would also match the shorter one.

Consider the following example:

```
[edit firewall family inet filter filter_on_src_addr term term1 from]
source-address {
   172.16.0.0/10;
   172.16.2.0/24 except;
   192.168.1.0;
   192.168.1.192/26 except;
   192.168.1.254;
   172.16.3.0/24; # ignored
   10.2.2.2 except; # ignored
}
```

Within the **source-address** match condition, two addresses are ignored. The **172.16.3.0/16** value is ignored because it falls under the address **172.16.0.0/10**, which is the same type. The **10.2.2.2 except** value is ignored because it is subsumed by the implicit **0.0.0.0/0 except** match value.

Suppose the following source IP address are evaluated by this firewall filter:

- Source IP address **172.16.1.2**—This address matches the **172.16.0.0/10** prefix, and thus the action in the **then** statement is taken.
- Source IP address **172.16.2.2**—This address matches the **172.16.2.0/24** prefix. Because this prefix is negated (that is, includes the **except** keyword), an explicit *mismatch* occurs. The next term in the filter is evaluated, if there is one. If there are no more terms, the packet is discarded.
- Source IP address **10.1.2.3**—This address does not match any of the prefixes included in the **source-address** condition. Instead, it matches the implicit **0.0.0.0/0 except** at the end of the list of prefixes configured under the **source-address** match condition, and is considered to be a mismatch.

  The **172.16.3.0/24** statement is ignored because it falls under the address **172.16.0.0/10**—both are the same type.

The **10.2.2.2 except** statement is ignored because it is subsumed by the implicit **0.0.0.0/0 except** statement at the end of the list of prefixes configured under the **source-address** match condition.

BEST PRACTICE: When a firewall filter term includes the **from address** *address* match condition and a subsequent term includes the **from source-address** *address* match condition for the same address, packets might be processed by the latter term before they are evaluated by any intervening terms. As a result, packets that should be rejected by the intervening terms might be accepted instead, or packets that should be accepted might be rejected instead.

To prevent this from occurring, we recommend that you do the following. For every firewall filter term that contains the **from address** *address* match condition, replace that term with two separate terms: one that contains the **from source-address** *address* match condition, and another that contains the **from destination-address** *address* match condition.

Matching an Address Field to a Prefix List

You can define a list of IPv4 or IPv6 address prefixes for use in a routing policy statement or in a stateless firewall filter match condition that evaluates packet address fields.

To define a list of IPv4 or IPv6 address prefixes, include the **prefix-list** ***prefix-list*** statement.

```
prefix-list name {
   ip-addresses;
   apply-path path;
}
```

You can include the statement at the following hierarchy levels:

- **[edit policy-options]**
- **[edit logical-systems** ***logical-system-name*** **policy-options]**

After you have defined a prefix list, you can use it when specifying a firewall filter match condition based on an IPv4 or IPv6 address prefix.

```
[edit firewall family family-name filter filter-name term term-name]
from {
   source-prefix-list {
      prefix-lists;
   }
   destination-prefix-list {
      prefix-lists;
   }
}
```

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 424
- Firewall Filter Match Conditions Based on Numbers or Text Aliases on page 458

Copyright © 2012, Juniper Networks, Inc.

- Firewall Filter Match Conditions Based on Bit-Field Values on page 453
- Firewall Filter Match Conditions Based on Address Classes on page 467

Firewall Filter Match Conditions Based on Address Classes

For IPv4 and IPv6 traffic only, you can use class-based firewall filter conditions to match packet fields based on source class or destination class.

- Source-Class Usage on page 467
- Destination-Class Usage on page 467
- Guidelines for Applying SCU or DCU Firewall Filters to Output Interfaces on page 467

Source-Class Usage

A *source class* is a set of source prefixes grouped together and given a class name. To configure a firewall filter term that matches an IP source address field to one or more source classes, use the **source-class *class-name*** match condition under the **[edit firewall family (inet | inet6) filter *filter-name* term *term-name* from]** hierarchy level.

Source-class usage (SCU) enables you to monitor the amount of traffic originating from a specific prefix. With this feature, usage can be tracked and customers can be billed for the traffic they receive.

Destination-Class Usage

A *destination class* is a set of destination prefixes grouped together and given a class name. To configure a firewall filter term that matches an IP destination address field to one or more destination classes, use the **destination-class *class-name*** match condition at the **[edit firewall family (inet | inet6) filter *filter-name* term *term-name* from]** hierarchy level.

Destination-class usage (DCU) enables you can track how much traffic is sent to a specific prefix in the core of the network originating from one of the specified interfaces.

Note, however, that DCU limits your ability to keep track of traffic moving in the reverse direction. It can account for all traffic that arrives on a core interface and heads toward a specific customer, but it cannot count traffic that arrives on a core interface from a specific prefix.

Guidelines for Applying SCU or DCU Firewall Filters to Output Interfaces

When applying a SCU or DCU firewall filter to an interface, keep the following guidelines in mind:

- Output interfaces—Class-based firewall filter match conditions work only for firewall filters that you apply to output interfaces. This is because the SCU and DCU are determined after route lookup occurs.
- Input interfaces—Although you can specify a source class and destination class for an input firewall filter, the counters are incremented only if the firewall filter is applied on the output interface.

Copyright © 2012, Juniper Networks, Inc.

- Output interfaces for tunnel traffic—SCU and DCU are not supported on the interfaces you configure as the output interface for tunnel traffic for transit packets exiting the router through the tunnel.

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 424
- Standard Firewall Filter Match Conditions for IPv4 Traffic on page 436
- Standard Firewall Filter Match Conditions for IPv6 Traffic on page 446
- *Junos OS Source Class Usage Feature Guide*
- Firewall Filter Match Conditions Based on Numbers or Text Aliases on page 458
- Firewall Filter Match Conditions Based on Bit-Field Values on page 453
- Firewall Filter Match Conditions Based on Address Fields on page 459

Standard Firewall Filter Terminating Actions

Standard stateless firewall filters support different sets of terminating actions for each protocol family.

NOTE: You cannot configure the next term action with a *terminating* action in the same filter term. However, you can configure the next term action with another *nonterminating* action in the same filter term.

Table 23 on page 468 describes the terminating actions you can specify in a standard firewall filter term.

Table 23: Terminating Actions for Standard Firewall Filters

Terminating Action	Description	Protocols
accept	Accept the packet.	• family any • family inet • family inet6 • family mpls • family vpls • family ccc • family bridge
discard	Discard a packet silently, without sending an Internet Control Message Protocol (ICMP) message. Discarded packets are available for logging and sampling.	• family any • family inet • family inet6 • family mpls • family vpls • family ccc • family bridge

Copyright © 2012, Juniper Networks, Inc.

Table 23: Terminating Actions for Standard Firewall Filters *(continued)*

Terminating Action	Description	Protocols
logical-system *logical-system-name*	Direct the packet to the specified logical system. NOTE: This action is not supported on PTX series packet transport switches.	• **family inet** • **family inet6**
reject *message-type*	Reject the packet and return an ICMPv4 or ICMPv6 message: • If no ***message-type*** is specified, a **destination unreachable** message is returned by default. • If **tcp-reset** is specified as the ***message-type***, **tcp-reset** is returned only if the packet is a TCP packet. Otherwise, the **administratively-prohibited** message, which has a value of 13, is returned. • If any other ***message-type*** is specified, that message is returned. NOTE: Rejected packets can be sampled or logged if you configure the **sample** or **syslog** action. The ***message-type*** can be one of the following values: **address-unreachable**, **administratively-prohibited**, **bad-host-tos**, **bad-network-tos**, **beyond-scope**, **fragmentation-needed**, **host-prohibited**, **host-unknown**, **host-unreachable**, **network-prohibited**, **network-unknown**, **network-unreachable**, **no-route**, **port-unreachable**, **precedence-cutoff**, **precedence-violation**, **protocol-unreachable**, **source-host-isolated**, **source-route-failed**, or **tcp-reset**.	• **family inet** • **family inet6**
routing-instance *routing-instance-name*	Direct the packet to the specified routing instance. NOTE: This action is not supported on PTX series packet transport switches.	• **family inet** • **family inet6**
topology *topology-name*	Direct the packet to the specified topology. NOTE: This action is not supported on PTX series packet transport switches.	• **family inet** • **family inet6**

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 424
- Standard Firewall Filter Nonterminating Actions on page 469

Standard Firewall Filter Nonterminating Actions

Standard stateless firewall filters support different sets of nonterminating actions for each protocol family.

NOTE: You cannot configure the **next term** action with a *terminating* action in the same filter term. However, you can configure the **next term** action with another *nonterminating* action in the same filter term.

Table 24 on page 470 describes the nonterminating actions you can configure for a standard firewall filter term.

Table 24: Nonterminating Actions for Standard Firewall Filters

Nonterminating Action	Description	Protocol Families
count *counter-name*	Count the packet in the named counter.	• family any • family inet • family inet6 • family mpls • family vpls • family ccc • family bridge

Copyright © 2012, Juniper Networks, Inc.

Table 24: Nonterminating Actions for Standard Firewall Filters *(continued)*

Nonterminating Action	Description	Protocol Families
dscp ***value***	Set the IPv4 Differentiated Services code point (DSCP) bit. You can specify a numerical value from **0** through **63**. To specify the value in hexadecimal form, include **0x** as a prefix. To specify the value in binary form, include **b** as a prefix. The default DSCP value is best effort, that is, **be** or **0**. You can also specify on the following text synonyms: • **af11**—Assured forwarding class 1, low drop precedence • **af12**—Assured forwarding class 1, medium drop precedence • **af13**—Assured forwarding class 1, high drop precedence • **af21**—Assured forwarding class 2, low drop precedence • **af22**—Assured forwarding class 2, medium drop precedence • **af23**—Assured forwarding class 2, high drop precedence • **af31**—Assured forwarding class 3, low drop precedence • **af32**—Assured forwarding class 3, medium drop precedence • **af33**—Assured forwarding class 3, high drop precedence • **af41**—Assured forwarding class 4, low drop precedence • **af42**—Assured forwarding class 4, medium drop precedence • **af43**—Assured forwarding class 4, high drop precedence • **be**—Best effort • **cs0**—Class selector 0 • **cs1**—Class selector 1 • **cs2**—Class selector 2 • **cs3**—Class selector 3 • **cs4**—Class selector 4 • **cs5**—Class selector 5 • **cs6**—Class selector 6 • **cs7**—Class selector 7 • **ef**—Expedited forwarding NOTE: This action is not supported on PTX Series packet transport switches. NOTE: The actions **dscp 0** or **dscp be** are supported only on T320, T640, T1600, TX Matrix, TX Matrix Plus,and M320 routers and on the 10-Gigabit Ethernet Modular Port Concentrators (MPC), 60-Gigabit Ethernet MPC, 60-Gigabit Ethernet Queuing MPC, and 60-Gigabit Ethernet Enhanced Queuing MPC on MX Series routers. However, these actions are not supported on Enhanced III Flexible PIC Concentrators (FPCs) on M320 routers. NOTE: On T4000 routers, the **dscp 0** action is not supported during the interoperation between a T1600 Enhanced Scaling Type 4 FPC and a T4000 Type 5 FPC.	**family inet**

Table 24: Nonterminating Actions for Standard Firewall Filters *(continued)*

Nonterminating Action	Description	Protocol Families
forwarding-class *class-name*	Classify the packet to the named forwarding class: • *forwarding-class-name* • **assured-forwarding** • **best-effort** • **expedited-forwarding** • **network-control**	• **family any** • **family inet** • **family inet6** • **family mpls** • **family vpls** • **family ccc** • **family bridge**
ipsec-sa *ipsec-sa*	Use the specified IPsec security association. NOTE: This action is not supported on MX Series routers, Type 5 FPCs on T4000 routers, and PTX Series packet transport switches.	**family inet**
load-balance *group-name*	Use the specified load-balancing group. NOTE: This action is not supported on MX Series routers or PTX Series packet transport switches.	**family inet**
log	Log the packet header information in a buffer within the Packet Forwarding Engine. You can access this information by issuing the **show firewall log** command at the command-line interface (CLI).	• **family inet** • **family inet6**
logical-system *logical-system-name*	Direct packets to a specific logical system.	• **family inet** • **family inet6**
loss-priority (high | medium-high | medium-low | low)	Set the packet loss priority (PLP) level. You cannot also configure the **three-color-policer** nonterminating action for the same firewall filter term. These two nonterminating actions are mutually exclusive. Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers. For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the **tri-color** statement at the **[edit class-of-service]** hierarchy level to commit a PLP configuration with any of the four levels specified. If the **tri-color** statement is not enabled, you can only configure the **high** and **low** levels. This applies to all protocol families. For information about the **tri-color** statement and for information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the *Junos OS Class of Service Configuration Guide*.	• **family any** • **family inet** • **family inet6** • **family mpls** • **family vpls** • **family ccc** • **family bridge**
next-hop-group *group-name*	Use the specified next-hop group.	**family inet**
packet-mode	Updates a bit field in the packet key buffer, which specifies traffic that will bypass flow-based forwarding. Packets with the **packet-mode** action modifier follow the packet-based forwarding path and bypass flow-based forwarding completely. For more information about selective stateless packet-based services, see the *Junos OS Security Configuration Guide*.	**family any**

Copyright © 2012, Juniper Networks, Inc.

Table 24: Nonterminating Actions for Standard Firewall Filters *(continued)*

Nonterminating Action	Description	Protocol Families
policer ***policer-name***	Name of policer to use to rate-limit traffic. NOTE: For IPv6, applies to SRX100, SRX210, SRX220, SRX240, and SRX650 devices only.	• **family any** • **family inet** • **family inet6** • **family mpls** • **family vpls** • **family ccc** • **family bridge**
port-mirror	Port-mirror the packet based on the specified family. Supported on M120 routers, M320 routers configured with Enhanced III FPCs, and MX Series routers only. NOTE: This action is not supported on T4000 Type 5 FPCs.	• **family inet** • **family inet6** • **family vpls** • **family ccc** • **family bridge**
prefix-action ***action-name***	Count or police packets based on the specified action name. NOTE: This action is not supported on PTX Series packet transport switches.	**family inet**
sample	Sample the packet. NOTE: The Junos OS does not sample packets originating from the router. If you configure a filter and apply it to the output side of an interface, then only the transit packets going through that interface are sampled. Packets that are sent from the Routing Engine to the Packet Forwarding Engine are not sampled. NOTE: This action is not supported on T4000 Type 5 FPCs.	• **family inet** • **family inet6** • **family mpls**
service-accounting	Count the packet for service accounting. The count is applied to a specific named counter (**__junos-dyn-service-counter**) that RADIUS can obtain. NOTE: This action is not supported on T4000 Type 5 FPCs and PTX Series packet transport switches. For more information, see "Configuring Service Packet Counting" in the *Junos OS Subscriber Access Configuration Guide*.	• **family inet** • **family inet6**
service-filter-hit	(Only if the **service-filter-hit** flag is marked by a previous filter in the current type of chained filters) Direct the packet to the next type of filters. Indicate to subsequent filters in the chain that the packet was already processed. This action, coupled with the **service-filter-hit** match condition in receiving filters, helps to streamline filter processing. NOTE: This action is not supported on T4000 Type 5 FPCs and PTX Series packet transport switches. For more information, see "Configuring Firewall Filter Bypass" in the *Junos OS Subscriber Access Configuration Guide*.	• **family inet** • **family inet6**

Copyright © 2012, Juniper Networks, Inc.

Table 24: Nonterminating Actions for Standard Firewall Filters *(continued)*

Nonterminating Action	Description	Protocol Families
syslog	Log the packet to the system log file.	• **family inet** • **family inet6**
three-color-policer (single-rate | two-rate) *policer-name*	Police the packet using the specified single-rate or two-rate three-color-policer. You cannot also configure the **loss-priority** action for the same firewall filter term. These two actions are mutually exclusive.	• **family inet** • **family inet6** • **family mpls** • **family vpls** • **family ccc** • **family bridge**
traffic-class *value*	Specify the traffic-class code point. You can specify a numerical value from **0** through **63**. To specify the value in hexadecimal form, include **0x** as a prefix. To specify the value in binary form, include **b** as a prefix. The default traffic-class value is best effort, that is, **be** or **0**. In place of the numeric value, you can specify one of the following text synonyms: • **af11**—Assured forwarding class 1, low drop precedence • **af12**—Assured forwarding class 1, medium drop precedence • **af13**—Assured forwarding class 1, high drop precedence • **af21**—Assured forwarding class 2, low drop precedence • **af22**—Assured forwarding class 2, medium drop precedence • **af23**—Assured forwarding class 2, high drop precedence • **af31**—Assured forwarding class 3, low drop precedence • **af32**—Assured forwarding class 3, medium drop precedence • **af33**—Assured forwarding class 3, high drop precedence • **af41**—Assured forwarding class 4, low drop precedence • **af42**—Assured forwarding class 4, medium drop precedence • **af43**—Assured forwarding class 4, high drop precedence • **be**—Best effort • **cs0**—Class selector 0 • **cs1**—Class selector 1 • **cs2**—Class selector 2 • **cs3**—Class selector 3 • **cs4**—Class selector 4 • **cs5**—Class selector 5 • **cs6**—Class selector 6 • **cs7**—Class selector 7 • **ef**—Expedited forwarding NOTE: The actions **traffic-class 0** or **traffic-class be** are supported only on T Series and M320 routers and on the 10-Gigabit Ethernet Modular Port Concentrator (MPC), 60-Gigabit Ethernet MPC, 60-Gigabit Ethernet Queuing MPC, and 60-Gigabit Ethernet Enhanced Queuing MPC on MX Series routers. However, these actions are not supported on Enhanced III Flexible PIC Concentrators (FPCs) on M320 routers.	**family inet6**

Copyright © 2012, Juniper Networks, Inc.

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 424
- Standard Firewall Filter Terminating Actions on page 468

Trusted Source and Flood Prevention Stateless Firewall Filters

- Understanding How to Use Standard Firewall Filters on page 475
- Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources on page 476
- Example: Configuring a Stateless Firewall Filter to Protect Against TCP and ICMP Floods on page 481

Understanding How to Use Standard Firewall Filters

This topic covers the following information:

- Using Standard Firewall Filters to Affect Local Packets on page 475
- Using Standard Firewall Filters to Affect Data Packets on page 476

Using Standard Firewall Filters to Affect Local Packets

On a router, you can configure one physical loopback interface, **lo0**, and one or more addresses on the interface. The loopback interface is the interface to the Routing Engine, which runs and monitors all the control protocols. The loopback interface carries local packets only. Standard firewall filters applied to the loopback interface affect the local packets destined for or transmitted from the Routing Engine.

NOTE: When you create an additional loopback interface, it is important to apply a filter to it so the Routing Engine is protected. We recommend that when you apply a filter to the loopback interface, you include the **apply-groups** statement. Doing so ensures that the filter is automatically inherited on every loopback interface, including **lo0** and other loopback interfaces.

Trusted Sources

The typical use of a standard stateless firewall filter is to protect the Routing Engine processes and resources from malicious or untrusted packets. To protect the processes and resources owned by the Routing Engine, you can use a standard stateless firewall filter that specifies which protocols and services, or applications, are allowed to reach the Routing Engine. Applying this type of filter to the loopback interface ensures that the local packets are from a trusted source and protects the processes running on the Routing Engine from an external attack.

Copyright © 2012, Juniper Networks, Inc.

Flood Prevention

You can create standard stateless firewall filters that limit certain TCP and ICMP traffic destined for the Routing Engine. A router without this kind of protection is vulnerable to TCP and ICMP flood attacks, which are also called denial-of-service (DoS) attacks. For example:

- A TCP flood attack of SYN packets initiating connection requests can overwhelm the device until it can no longer process legitimate connection requests, resulting in denial of service.
- An ICMP flood can overload the device with so many echo requests (ping requests) that it expends all its resources responding and can no longer process valid network traffic, also resulting in denial of service.

Applying the appropriate firewall filters to the Routing Engine protects against these types of attacks.

Using Standard Firewall Filters to Affect Data Packets

Standard firewall filters that you apply to your router's transit interfaces evaluate only the user data packets that transit the router from one interface directly to another as they are being forwarded from a source to a destination. To protect the network as a whole from unauthorized access and other threats at specific interfaces, you can apply firewall filters router transit interfaces .

Related Documentation

- How Standard Firewall Filters Evaluate Packets
- Guidelines for Configuring Standard Firewall Filters on page 424
- Guidelines for Applying Standard Firewall Filters on page 429

Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources

This example shows how to create a stateless firewall filter that protects the Routing Engine from traffic originating from untrusted sources.

- Requirements on page 476
- Overview on page 477
- Configuration on page 477
- Verification on page 479

Requirements

No special configuration beyond device initialization is required before configuring stateless firewall filters.

Copyright © 2012, Juniper Networks, Inc.

Overview

In this example, you create a stateless firewall filter called protect-RE that discards all traffic destined for the Routing Engine except SSH and BGP protocol packets from specified trusted sources. This example includes the following firewall filter terms:

- **ssh-term**—Accepts TCP packets with a source address of **192.168.122.0/24** and a destination port that specifies SSH.
- **bgp-term**—Accepts TCP packets with a source address of **10.2.1.0/24** and a destination port that specifies BGP.
- **discard-rest-term**—For all packets that are not accepted by **ssh-term** or **bgp-term**, creates a firewall filter log and system logging records, then discards all packets.

NOTE: You can move terms within the firewall filter using the insert command. See insert in the *Junos OS CLI User Guide*.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter protect-RE term ssh-term from source-address
  192.168.122.0/24
set firewall family inet filter protect-RE term ssh-term from protocol tcp
set firewall family inet filter protect-RE term ssh-term from destination-port ssh
set firewall family inet filter protect-RE term ssh-term then accept
set firewall family inet filter protect-RE term bgp-term from source-address 10.2.1.0/24
set firewall family inet filter protect-RE term bgp-term from protocol tcp
set firewall family inet filter protect-RE term bgp-term from destination-port bgp
set firewall family inet filter protect-RE term bgp-term then accept
set firewall family inet filter protect-RE term discard-rest-term then log
set firewall family inet filter protect-RE term discard-rest-term then syslog
set firewall family inet filter protect-RE term discard-rest-term then discard
set interfaces lo0 unit 0 family inet filter input protect-RE
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure the stateless firewall filter:

1. Create the stateless firewall filter.

   ```
   [edit]
   user@host# edit firewall family inet filter protect-RE
   ```

2. Create the first filter term.

   ```
   [edit firewall family inet filter protect-RE]
   ```

```
user@host# edit term ssh-term
```

3. Define the protocol, destination port, and source address match conditions for the term.

```
[edit firewall family inet filter protect-RE term ssh-term]
user@host# set from protocol tcp destination-port ssh source-address
  192.168.122.0/24
```

4. Define the actions for the term.

```
[edit firewall family inet filter protect-RE term ssh-term]
user@host# set then accept
```

5. Create the second filter term.

```
[edit firewall family inet filter protect-RE]
user@host# edit term bgp-term
```

6. Define the protocol, destination port, and source address match conditions for the term.

```
[edit firewall family inet filter protect-RE term bgp-term]
user@host# set from protocol tcp destination-port bgp source-address 10.2.1.0/24
```

7. Define the action for the term.

```
[edit firewall family inet filter protect-RE term bgp-term]
user@host# set then accept
```

8. Create the third filter term.

```
[edit firewall family inet filter protect-RE]
user@host# edit term discard-rest-term
```

9. Define the action for the term.

```
[edit firewall family inet filter protect-RE term discard-rest]
user@host# set then log syslog discard
```

10. Apply the filter to the input side of the Routing Engine interface.

```
[edit]
user@host# set interfaces lo0 unit 0 family inet filter input protect-RE
```

Results Confirm your configuration by entering the **show firewall** command and the **show interfaces lo0** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show firewall
family inet {
  filter protect-RE {
    term ssh-term {
      from {
        source-address {
          192.168.122.0/24;
        }
        protocol tcp;
        destination-port ssh;
```

 Copyright © 2012, Juniper Networks, Inc.

```
            }
            then accept;
        }
        term bgp-term {
            from {
                source-address {
                    10.2.1.0/24;
                }
                protocol tcp;
                destination-port bgp;
            }
            then accept;
        }
        term discard-rest-term {
            then {
                log;
                syslog;
                discard;
            }
        }
    }
}
```

```
user@host# show interfaces lo0
unit 0 {
    family inet {
        filter {
            input protect-RE;
        }
        address 127.0.0.1/32;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

```
[edit]
user@host# commit
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Displaying Stateless Firewall Filter Configurations on page 479
- Verifying a Services, Protocols, and Trusted Sources Firewall Filter on page 480
- Displaying Stateless Firewall Filter Logs on page 480

Displaying Stateless Firewall Filter Configurations

Purpose Verify the configuration of the firewall filter.

Action From configuration mode, enter the **show firewall** command and the **show interfaces lo0** command.

Meaning Verify that the output shows the intended configuration of the firewall filter. In addition, verify that the terms are listed in the order in which you want the packets to be tested. You can move terms within a firewall filter by using the **insert** CLI command.

Verifying a Services, Protocols, and Trusted Sources Firewall Filter

Purpose Verify that the actions of the firewall filter terms are taken.

Action Send packets to the device that match the terms. In addition, verify that the filter actions are *not* taken for packets that do not match.

- Use the **ssh** ***host-name*** command from a host at an IP address that matches **192.168.122.0/24** to verify that you can log in to the device using only SSH from a host with this address prefix.
- Use the **show route summary** command to verify that the routing table on the device does not contain any entries with a protocol other than **Direct**, **Local**, **BGP**, or **Static**.

Sample Output

```
% ssh 192.168.249.71
%ssh host
user@host's password:
--- JUNOS 6.4-20040518.0 (JSERIES) #0: 2004-05-18 09:27:50 UTC

user@host>

user@host> show route summary
Router ID: 192.168.249.71

inet.0: 34 destinations, 34 routes (33 active, 0 holddown, 1 hidden)
              Direct:     10 routes,      9 active
               Local:      9 routes,      9 active
                 BGP:     10 routes,     10 active
              Static:      5 routes,      5 active
...
```

Meaning Verify the following information:

- You can successfully log in to the device using SSH.
- The **show route summary** command does not display a protocol other than **Direct**, **Local**, **BGP**, or **Static**.

Displaying Stateless Firewall Filter Logs

Purpose Verify that packets are being logged. If you included the **log** or **syslog** action in a term, verify that packets matching the term are recorded in the firewall log or your system logging facility.

Action From operational mode, enter the **show firewall log** command.

Sample Output

```
user@host> show firewall log
```

 Copyright © 2012, Juniper Networks, Inc.

```
Log :
Time      Filter    Action Interface     Protocol Src Addr      Dest Addr
15:11:02  pfe       D      ge-0/0/0.0    TCP      172.17.28.19  192.168.70.71
15:11:01  pfe       D      ge-0/0/0.0    TCP      172.17.28.19  192.168.70.71
15:11:01  pfe       D      ge-0/0/0.0    TCP      172.17.28.19  192.168.70.71
15:11:01  pfe       D      ge-0/0/0.0    TCP      172.17.28.19  192.168.70.71
...
```

Meaning Each record of the output contains information about the logged packet. Verify the following information:

- Under **Time**, the time of day the packet was filtered is shown.
- The **Filter** output is always **pfe**.
- Under **Action**, the configured action of the term matches the action taken on the packet—**A** (accept), **D** (discard), **R** (reject).
- Under **Interface**, the inbound (ingress) interface on which the packet arrived is appropriate for the filter.
- Under **Protocol**, the protocol in the IP header of the packet is appropriate for the filter.
- Under **Src Addr**, the source address in the IP header of the packet is appropriate for the filter.
- Under **Dest Addr**, the destination address in the IP header of the packet is appropriate for the filter.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- show route summary in the *Junos OS Routing Protocols and Policies Command Reference*
- show firewall in the *Junos OS Routing Protocols and Policies Command Reference*
- show firewall log in the *Junos OS Routing Protocols and Policies Command Reference*
- show interfaces (Loopback) in the *Junos OS Interfaces Command Reference*

Example: Configuring a Stateless Firewall Filter to Protect Against TCP and ICMP Floods

This example shows how to create a stateless firewall filter that protects against TCP and ICMP denial-of-service attacks.

- Requirements on page 481
- Overview on page 482
- Configuration on page 482
- Verification on page 486

Requirements

No special configuration beyond device initialization is required before configuring stateless firewall filters.

Overview

In this example, you create a stateless firewall filter called **protect-RE** that polices TCP and ICMP packets. This example includes the following policers:

- **tcp-connection-policer**—Limits the traffic rate of the TCP packets to 500,000 bps and the burst size to 15,000 bytes. Packets that exceed the traffic rate are discarded.
- **icmp-policer**—Limits the traffic rate of the ICMP packets to 1,000,000 bps and the burst size to 15,000 bytes. Packets that exceed the traffic rate are discarded.

When specifying limits, the bandwidth limit can be from 32,000 bps to 32,000,000,000 bps and the burst-size limit can be from 1,500 bytes through 100,000,000 bytes. Use the following abbreviations when specifying limits: k (1,000), m (1,000,000), and g (1,000,000,000).

Each policer is incorporated into the action of a filter term. This example includes the following terms:

- **tcp-connection-term**—Polices certain TCP packets with a source address of 192.168.122.0/24 or 10.2.1.0/24. These addresses are defined in the **trusted-addresses** prefix list.

  Policed packets include connection request packets (SYN and ACK flag bits equal **1** and **0**), connection release packets (FIN flag bit equals **1**), and connection reset packets (RST flag bit equals **1**).
- **icmp-term**—Polices echo request packets, echo response packets, unreachable packets, and time-exceeded packets. All of these ICMP packets are counted in the **icmp-counter** counter.

NOTE: **You can move terms within the firewall filter by using the insert command. See insert in the** *Junos OS CLI User Guide*.

If you want to include the terms created in this procedure in the **protect-RE** firewall filter configured in "Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources" on page 476, perform the configuration tasks in this example first. Then configure the terms as described in "Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources" on page 476. This approach ensures that the rate-limiting terms are included as the first two terms in the firewall filter.

NOTE: **You can move terms within the firewall filter by using the insert command. See insert in the** *Junos OS CLI User Guide*.

Configuration

CLI Quick Configuration

To quickly configure the stateless firewall filter, copy the following commands to a text file, remove any line breaks, and then paste the commands into the CLI.

Copyright © 2012, Juniper Networks, Inc.

```
[edit]
set firewall family inet filter protect-RE term tcp-connection-term from source-prefix-list
   trusted-addresses
set firewall family inet filter protect-RE term tcp-connection-term from protocol tcp
set firewall family inet filter protect-RE term tcp-connection-term from tcp-flags "(syn
   & !ack) | fin | rst"
set firewall family inet filter protect-RE term tcp-connection-term then policer
   tcp-connection-policer
set firewall family inet filter protect-RE term tcp-connection-term then accept
set firewall family inet filter protect-RE term icmp-term from protocol icmp
set firewall family inet filter protect-RE term icmp-term from icmp-type echo-request
set firewall family inet filter protect-RE term icmp-term from icmp-type echo-reply
set firewall family inet filter protect-RE term icmp-term from icmp-type unreachable
set firewall family inet filter protect-RE term icmp-term from icmp-type time-exceeded
set firewall family inet filter protect-RE term icmp-term then policer icmp-policer
set firewall family inet filter protect-RE term icmp-term then count icmp-counter
set firewall family inet filter protect-RE term icmp-term then accept
set firewall policer tcp-connection-policer filter-specific
set firewall policer tcp-connection-policer if-exceeding bandwidth-limit 1m
set firewall policer tcp-connection-policer if-exceeding burst-size-limit 15k
set firewall policer tcp-connection-policer then discard
set firewall policer icmp-policer filter-specific
set firewall policer icmp-policer if-exceeding bandwidth-limit 1m
set firewall policer icmp-policer if-exceeding burst-size-limit 15k
set firewall policer icmp-policer then discard
set policy-options prefix-list trusted-addresses 10.2.1.0/24
set policy-options prefix-list trusted-addresses 192.168.122.0/24
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode.

To configure stateless firewall filter policers:

1. Define the first policer.

   ```
   [edit]
   user@host# edit firewall policer tcp-connection-policer
   ```

2. Define the action for the policer.

   ```
   [edit firewall policer tcp-connection-policer]
   user@host# set then discard
   ```

3. Define the rate limits for the policer.

   ```
   [edit firewall policer tcp-connection-policer]
   user@host# set filter-specific
   user@host# set if-exceeding burst-size-limit 15k bandwidth-limit 1m
   ```

4. Define the second policer.

   ```
   [edit]
   user@host# edit firewall policer icmp-policer
   ```

5. Define the action for the policer.

   ```
   [edit firewall policer icmp-policer]
   ```

```
user@host# set then discard
```

6. Set the rate limits for the policer.

```
[edit firewall policer icmp-policer]
user@host# set filter-specific
user@host# set if-exceeding burst-size-limit 15k bandwidth-limit 1m
```

7. Define the prefix list.

```
[edit]
user@host# set policy-options prefix-list trusted-addresses 192.168.122.0/24
user@host# set policy-options prefix-list trusted-addresses 10.2.1.0/24
```

8. Create the stateless firewall filter.

```
[edit]
user@host# edit firewall family inet filter protect-RE
```

9. Define the first term for the filter.

```
[edit firewall family inet filter protect-RE]
user@host# edit term tcp-connection-term
```

10. Define the source address match condition for the term.

```
[edit firewall family inet filter protect-RE term tcp-connection-term]
user@host# set from source-prefix-list trusted-addresses
```

11. Define protocol match conditions for the term.

```
[edit firewall family inet filter protect-RE term tcp-connection-term]
user@host# set from protocol tcp tcp-flags "(syn & !ack) | fin | rst"
```

12. Define the actions for the term.

```
[edit firewall family inet filter protect-RE term tcp-connection-term]
user@host# set then policer tcp-connection-policer accept
```

13. Define the second term.

```
[edit]
user@host# edit firewall family inet filter protect-RE term icmp-term
```

14. Define the protocol for the term.

```
[edit firewall family inet filter protect-RE term icmp-term]
user@host# set from protocol icmp
```

15. Define the match conditions for the term.

```
[edit firewall family inet filter protect-RE term icmp-term]
user@host# set from icmp-type [echo-request echo-reply unreachable
  time-exceeded]
```

16. Define the action for the term.

```
[edit firewall family inet filter protect-RE term icmp-term]
user@host# set then policer icmp-policer count icmp-counter accept
```


Copyright © 2012, Juniper Networks, Inc.

Results Confirm your configuration by entering the **show firewall** command and the **show policy-options** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show firewall
family inet {
   filter protect-RE {
      term tcp-connection-term {
         from {
            source-prefix-list {
               trusted-addresses;
            }
            protocol tcp;
            tcp-flags "(syn & !ack) | fin | rst";
         }
         then {
            policer tcp-connection-policer;
            accept;
         }
      }
      term icmp-term {
         from {
            protocol icmp;
            icmp-type [ echo-request echo-reply unreachable time-exceeded ];
         }
         then {
            policer icmp-policer;
            count icmp-counter;
            accept;
         }
      }
   }
}
policer tcp-connection-policer {
   filter-specific;
   if-exceeding {
      bandwidth-limit 1m;
      burst-size-limit 15k;
   }
   then discard;
}
policer icmp-policer {
   filter-specific;
   if-exceeding {
      bandwidth-limit 1m;
      burst-size-limit 15k;
   }
   then discard;
}
```

```
user@host# show policy-options
prefix-list trusted-addresses {
   10.2.1.0/24;
   192.168.122.0/24;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Displaying Stateless Firewall Filter Configurations on page 486
- Verifying a TCP and ICMP Flood Firewall Filter on page 486
- Displaying Firewall Filter Statistics on page 487

Displaying Stateless Firewall Filter Configurations

Purpose Verify the configuration of the firewall filter.

Action From configuration mode, enter the **show firewall** command.

Meaning Verify that the output shows the intended configuration of the firewall filter. In addition, verify that the terms are listed in the order in which you want the packets to be tested. You can move terms within a firewall filter by using the **insert** CLI command.

Verifying a TCP and ICMP Flood Firewall Filter

Purpose Verify that the actions of the firewall filter terms are taken.

Action Send packets to the device that match the terms. In addition, verify that the filter actions are *not* taken for packets that do not match.

- Verify that the device can establish only TCP sessions with a host at an IP address that matches 192.168.122.0/24 or 10.2.1.0/24. For example, log in to the device with the **telnet** ***host-name*** command from another host with one of these address prefixes.
- Use the **ping** ***host-name*** command to verify that the device responds only to ICMP packets (such as ping requests) that do not exceed the policer traffic rates.
- Use the **ping** ***host-name*** **size** ***bytes*** command to exceed the policer traffic rates by sending ping requests with large data payloads.

Sample Output

```
user@host> telnet 192.168.249.71
Trying 192.168.249.71...
Connected to host.acme.net.
Escape character is '^]'.

host (ttyp0)

login: user
Password:

--- JUNOS 6.4-20040521.1 built 2004-05-21 09:38:12 UTC

user@host>
```

 Copyright © 2012, Juniper Networks, Inc.

```
user@host> ping 192.168.249.71
PING host-ge-000.acme.net (192.168.249.71): 56 data bytes
64 bytes from 192.168.249.71: icmp_seq=0 ttl=253 time=11.946 ms
64 bytes from 192.168.249.71: icmp_seq=1 ttl=253 time=19.474 ms
64 bytes from 192.168.249.71: icmp_seq=2 ttl=253 time=14.639 ms
...

user@host> ping 192.168.249.71 size 20000
PING host-ge-000.acme.net (192.168.249.71): 20000 data bytes
^C
--- host-ge-000.acme.net ping statistics ---
12 packets transmitted, 0 packets received, 100% packet loss
```

Meaning Verify the following information:

- You can successfully log in to the device using Telnet.
- The device sends responses to the **ping host** command.
- The device does not send responses to the **ping host size 20000** command.

Displaying Firewall Filter Statistics

Purpose Verify that packets are being policed and counted.

Action From operational mode, enter the **show firewall filter** ***filter-name*** command.

Sample Output

```
user@host> show firewall filter protect-RE
Filter: protect-RE
Counters:
Name                                                Bytes              Packets
icmp-counter                                      1040000                 5600
Policers:
Name                                            Packets
tcp-connection-policer                        643254873
icmp-policer                                       7391
```

Meaning Verify the following information:

- Next to **Filter**, the name of the firewall filter is correct.
- Under **Counters**:
  - Under **Name**, the names of any counters configured in the firewall filter are correct.
  - Under **Bytes**, the number of bytes that match the filter term containing the **count** ***counter-name*** action are shown.
  - Under **Packets**, the number of packets that match the filter term containing the **count** ***counter-name*** action are shown.
- Under **Policers**:
  - Under **Name**, the names of any policers configured in the firewall filter are correct.

- Under **Packets**, the number of packets that match the conditions specified for the policer are shown.

Related Documentation

- "Two-Color Policer Configuration Overview" in the *Junos OS Firewall Filter and Policer Configuration Guide*.
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- show firewall in the *Junos OS Routing Protocols and Policies Command Reference*
- ping in the *Junos OS System Basics and Services Command Reference*.
- telnet in the *Junos OS System Basics and Services Command Reference*.

Fragment Handling Stateless Firewall Filters

- Firewall Filters That Handle Fragmented Packets Overview on page 488
- Example: Configuring a Stateless Firewall Filter to Handle Fragments on page 488
- Example: Configuring a Filter to Match on IPv6 Flags on page 493

Firewall Filters That Handle Fragmented Packets Overview

You can create stateless firewall filters that handle fragmented packets destined for the Routing Engine. By applying these policies to the Routing Engine, you protect against the use of IP fragmentation as a means to disguise TCP packets from a firewall filter.

For example, consider an IP packet that is fragmented into the smallest allowable fragment size of 8 bytes (a 20-byte IP header plus an 8-byte payload). If this IP packet carries a TCP packet, the first fragment (fragment offset of 0) that arrives at the device contains only the TCP source and destination ports (first 4 bytes), and the sequence number (next 4 bytes). The TCP flags, which are contained in the next 8 bytes of the TCP header, arrive in the second fragment (fragment offset of 1).

On all SRX Series and J Series devices, fragmented packets are not sampled correctly by the firewall filter. When file sampling, port-mirroring and CFLOW is applied on an interface in output direction, packets are sampled before fragmenting the packet and packet-capture captures packet after fragmentation.

See RFC 1858, *Security Considerations for IP Fragment Filtering*.

Related Documentation

- Understanding How to Use Standard Firewall Filters on page 475
- Example: Configuring a Stateless Firewall Filter to Handle Fragments on page 488

Example: Configuring a Stateless Firewall Filter to Handle Fragments

This example shows how to create a stateless firewall filter that handles packet fragments.

- Requirements on page 489
- Overview on page 489

Copyright © 2012, Juniper Networks, Inc.

- Configuration on page 489
- Verification on page 492

Requirements

No special configuration beyond device initialization is required before configuring stateless firewall filters.

Overview

In this example, you create a stateless firewall filter called **fragment-RE** that accepts fragmented packets originating from 10.2.1.0/24 and destined for the BGP port. This example includes the following firewall filter terms:

- **not-from-prefix-term**-–Discards packets that are not from 10.2.1.0/24 to ensure that subsequent terms in the firewall filter are matched against packets from 10.2.1.0/24 only.
- **small-offset-term**—Discards small (1–5) offset packets to ensure that subsequent terms in the firewall filter can be matched against all the headers in the packet. In addition, the term adds a record to the system logging destinations for the firewall facility.
- **not-fragmented-term**—Accepts unfragmented TCP packets with a destination port that specifies the BGP protocol. A packet is considered unfragmented if the MF flag is not set and the fragment offset equals 0.
- **first-fragment-term**—Accepts the first fragment of a fragmented TCP packet with a destination port that specifies the BGP protocol.
- **fragment-term**—Accepts all fragments that were not discarded by **small-offset-term**. (packet fragments 6–8191). However, only those fragments that are part of a packet containing a first fragment accepted by **first-fragment-term** are reassembled by the destination device.

Packet fragments offset can be from 1 through 8191.

NOTE: You can move terms within the firewall filter by using the insert command. For more information, see “insert” in the *Junos OS CLI User Guide*.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter fragment-RE term not-from-prefix-term from source-address
  0.0.0.0/0
set firewall family inet filter fragment-RE term not-from-prefix-term from source-address
  10.2.1.0/24 except
set firewall family inet filter fragment-RE term not-from-prefix-term then discard
```

```
set firewall family inet filter fragment-RE term small-offset-term from fragment-offset
  1-5
set firewall family inet filter fragment-RE term small-offset-term then syslog
set firewall family inet filter fragment-RE term small-offset-term then discard
set firewall family inet filter fragment-RE term not-fragmented-term from fragment-offset
  0
set firewall family inet filter fragment-RE term not-fragmented-term from fragment-flags
  "!more-fragments"
set firewall family inet filter fragment-RE term not-fragmented-term from protocol tcp
set firewall family inet filter fragment-RE term not-fragmented-term from destination-port
  bgp
set firewall family inet filter fragment-RE term not-fragmented-term then accept
set firewall family inet filter fragment-RE term first-fragment-term from first-fragment
set firewall family inet filter fragment-RE term first-fragment-term from protocol tcp
set firewall family inet filter fragment-RE term first-fragment-term from destination-port
  bgp
set firewall family inet filter fragment-RE term first-fragment-term then accept
set firewall family inet filter fragment-RE term fragment-term from fragment-offset
  6-8191
set firewall family inet filter fragment-RE term fragment-term then accept
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure the stateless firewall filter:

1. Define the stateless firewall filter.

   ```
   [edit]
   user@host# edit firewall family inet filter fragment-RE
   ```

2. Configure the first term for the filter.

   ```
   [edit firewall family inet filter fragment-RE ]
   user@host# set term not-from-prefix-term from source-address 0.0.0.0/0
   user@host# set term not-from-prefix-term from source-address 10.2.1.0/24 except
   user@host# set term not-from-prefix-term then discard
   ```

3. Define the second term for the filter.

   ```
   [edit firewall family inet filter fragment-RE]
   user@host# edit term small-offset-term
   ```

4. Define the match conditions for the term.

   ```
   [edit firewall family inet filter fragment-RE term small-offset-term]
   user@host# set from fragment-offset 1-5
   ```

5. Define the action for the term.

   ```
   [edit firewall family inet filter fragment-RE term small-offset-term]
   user@host# set then syslog discard
   ```

6. Define the third term for the filter.

   ```
   [edit]
   user@host# edit firewall family inet filter fragment-RE term not-fragmented-term
   ```


Copyright © 2012, Juniper Networks, Inc.

7. Define the match conditions for the term.

```
[edit firewall family inet filter fragment-RE term not-fragmented-term]
user@host# set from fragment-flags "!more-fragments" fragment-offset 0 protocol
  tcp destination-port bgp
```

8. Define the action for the term.

```
[edit firewall family inet filter fragment-RE term not-fragmented-term]
user@host# set then accept
```

9. Define the fourth term for the filter.

```
[edit]
user@host# edit firewall family inet filter fragment-RE term first-fragment-term
```

10. Define the match conditions for the term.

```
[edit firewall family inet filter fragment-RE term first-fragment-term]
user@host# set from first-fragment protocol tcp destination-port bgp
```

11. Define the action for the term.

```
[edit firewall family inet filter fragment-RE term first-fragment-term]
user@host# set then accept
```

12. Define the last term for the filter.

```
[edit]
user@host# edit firewall family inet filter fragment-RE term fragment-term
```

13. Define the match conditions for the term.

```
[edit firewall family inet filter fragment-RE term fragment-term]
user@host# set from fragment-offset 6–8191
```

14. Define the action for the term.

```
[edit firewall family inet filter fragment-RE term fragment-term]
user@host# set then accept
```

Results Confirm your configuration by entering the **show firewall** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show firewall
family inet {
  filter fragment-RE {
    term not-from-prefix-term {
      from {
        source-address {
          0.0.0.0/0;
          10.2.1.0/24 except;
        }
      }
      then discard;
    }
    term small-offset-term {
      from {
```

```
            fragment-offset 1-5;
          }
          then {
            syslog;
            discard;
          }
        }
        term not-fragmented-term {
          from {
            fragment-offset 0;
            fragment-flags "!more-fragments";
            protocol tcp;
            destination-port bgp;
          }
          then accept;
        }
        term first-fragment-term {
          from {
            first-fragment;
            protocol tcp;
            destination-port bgp;
          }
          then accept;
        }
        term fragment-term {
          from {
            fragment-offset 6-8191;
          }
          then accept;
        }
      }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Displaying Stateless Firewall Filter Configurations on page 492
- Verifying a Firewall Filter that Handles Fragments on page 493

Displaying Stateless Firewall Filter Configurations

Purpose Verify the configuration of the firewall filter. You can analyze the flow of the filter terms by displaying the entire configuration.

Action From configuration mode, enter the **show firewall** command.

Meaning Verify that the output shows the intended configuration of the firewall filter. In addition, verify that the terms are listed in the order in which you want the packets to be tested. You can move terms within a firewall filter by using the **insert** CLI command.

 Copyright © 2012, Juniper Networks, Inc.

Verifying a Firewall Filter that Handles Fragments

Purpose Verify that the actions of the firewall filter terms are taken.

Action Send packets to the device that match the terms.

Meaning Verify that packets from 10.2.1.0/24 with small fragment offsets are recorded in the device's system logging destinations for the firewall facility.

Related Documentation

- show route summary in the *Junos OS Routing Protocols and Policies Command Reference*.

Example: Configuring a Filter to Match on IPv6 Flags

This example shows how to configure a filter to match on IPv6 TCP flags.

- Requirements on page 493
- Overview on page 493
- Configuration on page 493
- Verification on page 494

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you configure a filter to match on IPv6 TCP flags. You can use this example to configure IPv6 TCP flags in the SRX100, SRX210, SRX240, SRX650, and J Series security devices and in M Series, MX Series, and T Series routing devices.

Configuration

Step-by-Step Procedure To configure a filter to match on IPv6 TCP flags:

1. Include the family statement at the firewall hierarchy level, specifying **inet6** as the protocol family.

   ```
   [edit]
   user@host# edit firewall family inet6
   ```

2. Create the stateless firewall filter.

   ```
   [edit firewall family inet6]
   user@host# edit filter tcpfilt
   ```

3. Define the first term for the filter.

   ```
   [edit firewall family inet6 filter tcpfilt]
   user@host# edit term 1
   ```

4. Define the source address match conditions for the term.

   [edit firewall family inet6 filter tcpfilt term 1]
   user@host# **set from next-header tcp tcp-flags syn**

5. Define the actions for the term.

   [edit firewall family inet6 filter tcpfilt term 1]
   user@host# **set then count tcp_syn_pkt log accept**

6. If you are done configuring the device, commit the configuration.

   [edit firewall family inet6 filter tcpfilt term 1]
   user@host **top**

   [edit]
   user@host# **commit**

Verification

To confirm that the configuration is working properly, enter the **show firewall filter tcpfilt** command.

Copyright © 2012, Juniper Networks, Inc.

PART 3

Debugging and Trace Operations

- Tracing Global Routing Protocol Operations on page 497

Copyright © 2012, Juniper Networks, Inc.

CHAPTER 16

Tracing Global Routing Protocol Operations

- Understanding Global Routing Protocol Tracing Operations on page 497
- Example: Tracing Global Routing Protocol Operations on page 498

Understanding Global Routing Protocol Tracing Operations

Global routing protocol tracing operations track all general routing operations and record them in a log file. To set protocol-specific tracing operations and to modify the global tracing operations for an individual protocol, configure tracing for that protocol.

Using the **traceoptions** statement, you can specify the following global routing protocol tracing flags:

- **all**—All tracing operations
- **condition-manager**—Condition manager events
- **config-internal**—Configuration internals
- **general**—All normal operations and routing table changes (a combination of the normal and route trace operations)
- **graceful-restart**—Graceful restart operations
- **normal**—All normal operations
- **nsr-synchronization**—Nonstop routing synchronization events
- **parse**—Configuration parsing
- **policy**—Policy operations and actions
- **regex-parse**—Regular expression parsing
- **route**—Routing table changes
- **state**—State transitions
- **task**—Interface transactions and processing
- **timer**—Timer usage

Copyright © 2012, Juniper Networks, Inc.

NOTE: Use the **all** flag with caution. This flag might cause the CPU to become very busy.

Related Documentation

- Example: Tracing Global Routing Protocol Operations on page 498
- *Junos OS System Basics Configuration Guide*

Example: Tracing Global Routing Protocol Operations

This example shows how to list and view files that are created when you enable global routing trace operations.

- Requirements on page 498
- Overview on page 498
- Configuration on page 499
- Verification on page 501

Requirements

You must have the **view** privilege.

Overview

To configure global routing protocol tracing, include the **traceoptions** statement at the **[edit routing-options]** hierarchy level:

```
traceoptions {
   file filename <files number> <size size> <world-readable | no-world-readable>;
   flag flag <disable>;
}
```

The flags in a **traceoptions flag** statement are identifiers. When you use the **set** command to configure a flag, any flags that might already be set are not modified. In the following example, setting the **timer** tracing flag has no effect on the already configured **task** flag. Use the **delete** command to delete a particular flag.

```
[edit routing-options traceoptions]
user@host# show
flag task;
user@host# set traceoptions flag timer
user@host# show
flag task;
flag timer;
user@host# delete traceoptions flag task
user@host# show
flag timer;
```

This example shows how to configure and view a trace file that tracks changes in the routing table. The steps can be adapted to apply to trace operations for any Junos OS hierarchy level that supports trace operations.

Copyright © 2012, Juniper Networks, Inc.

TIP: To view a list of hierarchy levels that support tracing operations, enter the **help apropos traceoptions** command in configuration mode.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set routing-options traceoptions file routing-table-changes
set routing-options traceoptions file size 10m
set routing-options traceoptions file files 10
set routing-options traceoptions flag route
set routing-options static route 1.1.1.2/32 next-hop 10.0.45.6
```

Configuring Trace Operations

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the *Junos OS CLI User Guide*.

To configure the trace operations:

1. Configure trace operations.

   ```
   [edit routing-options traceoptions]
   user@host# set file routing-table-changes
   user@host# set file size 10m
   user@host# set file files 10
   user@host# set flag route
   ```

2. Configure a static route to cause a change in the routing table.

   ```
   [edit routing-options static]
   user@host# set route 1.1.1.2/32 next-hop 10.0.45.6
   ```

3. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Copyright © 2012, Juniper Networks, Inc.

Viewing the Trace File

Step-by-Step Procedure

To view the trace file:

1. In operational mode, list the log files on the system.

```
user@host> file list /var/log

/var/log:
...
routing-table-changes
...
```

2. View the contents of the **routing-table-changes** file.

```
user@host> file show /var/log/routing-table-changes

Dec 15 11:09:29 trace_on: Tracing to "/var/log/routing-table-changes"
started
Dec 15 11:09:29.496507
Dec 15 11:09:29.496507 Tracing flags enabled: route
Dec 15 11:09:29.496507
Dec 15 11:09:29.533203 inet_routerid_notify: Router ID: 192.168.4.1
Dec 15 11:09:29.533334 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.533381 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.533420 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.534915 inet_routerid_notify: Router ID: 192.168.4.1
Dec 15 11:09:29.542934 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.549253 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.556878 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.582990 rt_static_reinit: examined 3 static nexthops, 0
unreferenced
Dec 15 11:09:29.589920
Dec 15 11:09:29.589920 task_reconfigure reinitializing done
...
```

3. Filter the output of the log file.

```
user@host> file show /var/log/routing-table-changes | match 1.1.1.2

Dec 15 11:15:30.780314 ADD      1.1.1.2/32          nhid 0 gw 10.0.45.6
      Static   pref 5/0 metric  at-0/2/0.0 <ctive Int Ext>
Dec 15 11:15:30.782276 KRT Request: send len 216 v104 seq 0 ADD route/user
 af 2 table 0 infot 0 addr 1.1.1.2 nhop-type unicast nhindex 663
```

4. View the tracing operations in real time by running the **monitor start** command with an optional **match** condition.

```
user@host> monitor start routing-table-changes | match 1.1.1.2

Aug 10 19:21:40.773467 BGP RECV         0.0.0.0/0
Aug 10 19:21:40.773685 bgp_rcv_nlri: 0.0.0.0/0
Aug 10 19:21:40.773778 bgp_rcv_nlri: 0.0.0.0/0 belongs to meshgroup
Aug 10 19:21:40.773832 bgp_rcv_nlri: 0.0.0.0/0 qualified bnp->ribact 0x0
 l2afcb 0x0
```

5. Deactivate the static route.

```
user@host# deactivate routing-options static route 1.1.1.2/32
user@host# commit
```


Copyright © 2012, Juniper Networks, Inc.

```
*** routing-table-changes ***
Dec 15 11:42:59.355557 CHANGE   1.1.1.2/32         nhid 663 gw 10.0.45.6
       Static   pref 5/0 metric  at-0/2/0.0 <Delete Int Ext>
Dec 15 11:42:59.426887 KRT Request: send len 216 v104 seq 0 DELETE
route/user af 2 table 0 infot 0 addr 1.1.1.2 nhop-type discard filtidx 0
Dec 15 11:42:59.427366 RELEASE  1.1.1.2/32         nhid 663 gw 10.0.45.6
       Static   pref 5/0 metric  at-0/2/0.0 <Release Delete Int Ext>
```

6. Halt the **monitor** command by pressing Enter and typing **monitor stop**.

```
[Enter]

user@host> monitor stop
```

7. When you are finished troubleshooting, consider deactivating trace logging to avoid any unnecessary impact to system resources.

   When configuration is deactivated, it appears in the configuration with the **inactive** tag.

```
[edit routing-options]
user@host# deactivate traceoptions
user@host# commit
```

```
[edit routing-options]

user@host# show


inactive: traceoptions {
    file routing-table-changes size 10m files 10;
    flag route;
}
static {
    inactive: route 1.1.1.2/32 next-hop 10.0.45.6;
}
```

8. To reactivate trace operations, use the **activate** configuration-mode statement.

```
[edit routing-options]
user@host# activate traceoptions
user@host# commit
```

Results From configuration mode, confirm your configuration by entering the **show routing-options** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show routing-options
traceoptions {
   file routing-table-changes size 10m files 10;
   flag route;
}
static {
   route 1.1.1.2/32 next-hop 10.0.45.6;
}
```

Verification

Confirm that the configuration is working properly.

Verifying That the Trace Log File Is Operating

Purpose Make sure that events are being written to the log file.

Action

```
user@host> show log routing-table-changes
Dec 15 11:09:29 trace_on: Tracing to "/var/log/routing-table-changes" started
```

Related Documentation

- Understanding Global Routing Protocol Tracing Operations on page 497
- *Junos OS System Basics and Services Command Reference*

 Copyright © 2012, Juniper Networks, Inc.

PART 4

Index

- Index on page 505

Copyright © 2012, Juniper Networks, Inc.

Index

Symbols

! (negation)
in firewall filters
bit-field logical operator....453
#, comments in configuration statements....xix
&, bit-field logical operator....453
( ), in syntax descriptions....xix
*,G notation, for multicast forwarding states....355
+
bit-field logical operator....453
, (comma), bit-field logical operator....453
< >, in syntax descriptions....xix
[ ], in configuration statements....xix
{ }, in configuration statements....xix
| (pipe)
in firewall filters
bit-field logical operator....453
| (pipe), in syntax descriptions....xix

A

ABRs *See* area border routers
actions
default, routing policy....390
final, routing policy....390
route list match types....397
routing policy....394
routing policy, summary of....394
actions, flow control....427
actions, nonterminating
for standard stateless firewall filters....469
actions, terminating
for standard stateless firewall filters....468
activate OSPF....165
active aggregate routes....91
active routes....265
active routes, versus passive routes....53
active statement
aggregate routes
usage guidelines....98
add-path statement
BGP
usage guidelines....269
address class, source or destination
stateless firewall filter match conditions
IPv4 traffic....436
IPv6 traffic....446
overview....467
address, source or destination
stateless firewall filter match conditions
IPv4 traffic....436
IPv6 traffic....446
addresses....216
IS-IS NETs....216
See also NETs
IS-IS NSAP addresses....216
multicast ranges....355
See also IPv4 addressing; IPv6 addressing
adjacencies, IS-IS
hello PDUs....217
See also IS-IS
administrative scoping....356
advertisements *See* LSAs; route advertisements
aggregate routes....91
aggregate statement
usage guidelines....91
aggregation, route....7
always compare, BGP MED option....302
always-compare-med option....265
ampersand (&), bit-field logical operator....453
area border routers
backbone area *See* backbone area
description....170
area statement
usage guidelines, backbone....172
usage guidelines, multiarea....174
areas *See* area border routers; backbone area; NSSAs; stub areas
AS path
ignoring in route selection....293
prepending....406
as-path statement
aggregate routes
usage guidelines....96
as-path-ignore
usage guidelines....265, 293
ASs
paths....229
aggregate routes....96

ASs (autonomous systems)
area border routers....170
breaking into confederations....321
description....4
IS-IS networks....215
stub areas *See* stub areas
authentication
BGP....231
IPsec
OSPFv2....187, 196
OSPFv3 196
MD5
multiple keys....193
OSPFv2....187
single key....191
OSPFv2....187
RIPv2, MD5....149
RIPv2, plain-text passwords....149
simple
OSPFv2....187
authentication configuration
BFD....346
Auto-RP....357
autonomous systems *See* ASs

B

backbone area
configuring....172
description....170
bandwidth-based metrics
OSPF....203
BFD
authentication configuration....346
protocol....329
traceoptions statement
usage guidelines....332
bfd-liveness-detection statement
static routes
usage guidelines....329
usage guidelines....338
BGP
advertising multiple paths to a destination....269, 292
ASs *See* ASs
authentication....231
communities
aggregate routes....95
description....254
external (EBGP)....229
groups....246, 301
hold time....231
identifier....231
ignoring the AS path attribute in route selection....293
injecting OSPF routes into BGP....402
internal (IBGP)....229
IP address....231
keepalive messages....232
local address....254
local preference....233
messages....230
neighbors BGP, peers *See* BGP, peers
NLRI....231
open messages....231
overview....228
path attributes....229, 231
peers....229, 301
point-to-point peer session (configuration editor)....247
routes....229
TCP....228
update messages....231
version supported....228
BGP (Border Gateway Protocol)
confederations *See* BGP confederations
internal peer session (configuration editor)....254
peering sessions *See* BGP peers; BGP sessions
policy to make routes less preferable....406
requirements....232
route reflectors *See* BGP route reflectors
route-flap damping....408
BGP confederations
creating (configuration editor)....322
description....321
route-flap damping....408
BGP groups
confederations (configuration editor)....322
BGP peers
external (configuration editor)....247
internal (configuration editor)....254
point-to-point connections....246
BGP route reflectors
cluster of clusters....305
creating (configuration editor)....306
description....304
multiple clusters....305

Copyright © 2012, Juniper Networks, Inc.

BGP sessions
internal (configuration editor).....254
point-to-point external (configuration editor).....247
sample peering session.....246
bit-field
logical operators.....453
bootstrap router.....357
Border Gateway Protocol *See* BGP
braces, in configuration statements.....xix
brackets
angle, in syntax descriptions.....xix
square, in configuration statements.....xix
branches.....354
See also multicast
brief statement
aggregate routes
usage guidelines.....97
BSR (bootstrap router).....357

C

Cisco non-deterministic, BGP MED option.....302
cisco-non-deterministic option.....265
CLNS (Connectionless Network Service) VPNs
static routes (without IS-IS).....79
cluster statement
usage guidelines.....306
clusters *See* BGP route reflectors
color statement
aggregate routes
usage guidelines.....94
comments, in configuration statements.....xix
communities
aggregate routes.....95
community statement
aggregate routes
usage guidelines.....95
complete sequence number PDU (CSNP).....218
confederation statement
usage guidelines.....322
confederations *See* BGP confederations
connectivity
unidirectional (RIP).....134
contributing routes
aggregate routes.....91
conventions
notice icons.....xviii
text and syntax.....xviii
CSNP (complete sequence number PDU).....218
curly braces, in configuration statements.....xix
customer support.....xx
contacting JTAC.....xx

D

damping statement
usage guidelines.....409
default-lsa statement
usage guidelines.....182
default-metric statement
usage guidelines.....178, 182
defaults
routing policy actions.....390
defaults statement
aggregate statement
usage guidelines.....91
static statement
usage guidelines.....24
demand-circuit statement
RIP
usage guidelines.....153
denial-of-service attacks, preventing.....481
dense routing mode, caution for use.....355
See also multicast routing modes
description statement
usage guidelines.....254
designated router
configuring.....168
controlling election.....168
OSPF.....166
designated router, IS-IS
about.....224
configuring election priority.....224
designated router, stopping outgoing PIM register messages on.....374
diagnosis
displaying stateless firewall filter configurations.....479, 486, 492
displaying stateless firewall filter statistics.....487
displaying static routes in the routing table.....89
verifying firewall filter handles fragments.....493
verifying multicast IGMP versions.....382
verifying multicast SAP and SDP configuration.....382
verifying OSPF host reachability.....213
verifying OSPF neighbors.....211
verifying OSPF routes.....211

Copyright © 2012, Juniper Networks, Inc.

verifying OSPF-enabled interfaces....210
verifying PIM mode and interface configuration....383
verifying PIM RPF routing table....384
verifying PIM RPs....383
verifying RIP host reachability 143, 152
verifying RIP message exchange....150
verifying RIP-enabled interfaces....151
verifying stateless firewall filter actions....480
verifying stateless firewall filter DoS protection....486
verifying stateless firewall filter flood protection....486
verifying stateless firewall filters with packet logs....480
discard statement, in static statement
usage guidelines....24
Distance Vector Multicast Routing Protocol....357
distance-vector routing protocols....131
See also RIP
documentation
comments on....xx
DoS (denial-of-service) attacks, preventing....481
downstream interfaces....354
See also multicast
DR *See* designated router
DSCP code point
stateless firewall filter match condition
IPv4 traffic....436
DVMRP (Distance Vector Multicast Routing Protocol)....357
dynamic routing....6

E

EBGP *See* BGP
EBGP (external BGP)
route-flap damping....408
ECMP
flow-based forwarding....412, 413
EGPs (exterior gateway protocols)....4
equal-cost multi-path *See* ECMP
Ethernet interfaces, unnumbered
as next-hop interface for static routes....47
exact route list match type....397
exclamation point (!), bit-field logical operator....453
export statement
forwarding table
usage guidelines....18
export statement, for routing policies....389
exterior gateway protocols....4
external-preference statement
OSPF
usage guidelines....208
external-router-id option....265

F

fault tolerance
advertising multiple paths to a destination....269, 292
firewall filters
verifying fragment handling....493
flap damping....408
parameters....409
flooding, preventing....481
flow control, actions in routing policies....394
flow-based forwarding
ECMP....412, 413
font conventions....xviii
forwarding class
stateless firewall filter match conditions
IPv4 traffic....436
IPv6 traffic....446
forwarding classes
policy to group source and destination prefixes....399
forwarding states, multicast notation....355
forwarding table
aggregate routes....97, 98
controlling static routes in....52, 53
description....5
static routes....24, 98
from statement, routing policy match conditions....392
full mesh requirement
fulfilling with confederations....321
fulfilling with route reflectors....304
full statement
aggregate routes
usage guidelines....97

G

groups
OSPF areas....174
RIP routers....138

H

handling packet fragments....488

Copyright © 2012, Juniper Networks, Inc.

hardware
supported platforms....xviii
hello PDUs....217
hop count, maximizing....132
See also RIP
host reachability
verifying for an OSPF network....213
verifying for RIP network hosts....143, 152
hostname
IS-IS identifier-to-hostname mapping....216

I

IBGP *See* BGP
IBGP (internal BGP)
full mesh (configuration editor)....246
ICMP (Internet Control Message Protocol), policers....481
identifiers
BGP *See* BGP, identifier
IGMP (Internet Group Management Protocol)
IGMPv1....357
IGMPv2....357
IGMPv3....357
setting the version....362
verifying the version....382
IGP plus MED, BGP option....302
IGPs (interior gateway protocols)....4
import statement, for routing policies....389
incoming metric (RIP)
description....144
indirect next hop....106
indirect-next-hop statement
usage guidelines....106
inet routing table....378
inet.2 routing table....15
inet6.0 routing table....15
instances
routing, multiple....11
interface statement
static routes
usage guidelines....47
interface-routes statement
usage guidelines....18
interfaces
configuring on logical systems....33
interior gateway protocols....4
Internet Control Message Protocol policers....481
Internet Group Management Protocol *See* IGMP
IP addresses
as IS-IS system identifiers....216
IPsec authentication
OSPFv2....187
IPsec security associations
OSPFv2....187
ipsec-sa statement
usage guidelines....196
IPv4 traffic
match conditions
standard stateless firewall filters....436
IPv6
static routes....28
IPv6 traffic
match conditions
standard stateless firewall filters....446
IS-IS (Intermediate System-to-Intermediate System)
about designated routers....224
adjacency establishment with hello PDUs....217
areas....215
ASs....215
configuring....219
configuring designated router election priority....224
CSNPs....218
hello PDUs....217
LSPs....218
NETs....216
See also NETs
NSAP addresses....216
overview....215
path selection....217
PSNPs....218
system identifiers....216
See also system identifiers
ISO network addresses, for IS-IS routers....216

K

keepalive messages....232

L

leaves....354
See also multicast
Level 1 areas, IS-IS....215
Level 2 areas, IS-IS....215
link-state PDUs *See* LSPs

load balancing
advertising multiple paths to a destination....269, 292
local-address statement
BFD
usage guidelines....329
usage guidelines....254
local-preference statement
usage guidelines....233
logical systems
configuring interfaces....33
configuring static routes on....35
connecting to a physical router....44
connecting within the same router....40
logical tunnel interfaces
configuring on logical systems....40
longer route list match type....397
loopback interface, applying stateless firewall filters to (configuration editor)....481
loss priority
stateless firewall filter match conditions
IPv4 traffic....436
IPv6 traffic....446
LSPs (link-state PDUs)
CSNPs....218
overview....218
PSNPs....218

M

MAC (media access control) addresses
as IS-IS system identifiers....216
manuals
comments on....xx
martian addresses....126
match condition categories
stateless firewall filters
matching on address classes....467
matching on address prefixes....459
matching on bit-field values....453
matching on numeric values....458
matching on text strings....458
match conditions
routing policy....392
routing policy, summary of....393
match conditions for standard stateless firewall filters
IPv4 traffic....436
IPv6 traffic....446
match types....397
maximum hop count, RIP....132
MD5 authentication
multiple keys
configuring....193
single key
configuring....191
understanding....187
md5 statement
usage guidelines
multiple keys....193
single key....191
MED (multiple exit discriminator)
always compare option....302
Cisco non-deterministic option....302
plus IGP option....302
med-plus-igp statement
usage guidelines....265
members statement
usage guidelines....322
metric statement
aggregate routes
usage guidelines....94
CLNS
usage guidelines....79
OSPF
usage guidelines....204
qualified next hop
usage guidelines....47
metrics
OSPF....202
minimum-interval statement
BFD
usage guidelines....329
minimum-receive-interval statement
BFD
usage guidelines....329
minimum-receive-ttl statement
BFD (static routes)
usage guidelines....329
MP-BGP....15
MSDP (Multicast Source Discovery Protocol)....357
multiarea network....174
multicast
*,G notation....355
administrative scoping....356
architecture....354
Auto-RP....357
BSR....357
downstream interface....354

Copyright © 2012, Juniper Networks, Inc.

DVMRP....357
forwarding state notation....355
IGMP *See* IGMP
IP address ranges....355
MSDP....357
network elements....355
overview....353
PGM....357
PIM dense mode *See* PIM
PIM register messages *See* PIM register messages
PIM source-specific multicast (SSM)....357
PIM sparse mode *See* PIM
preparation....359
preventing routing loops....356
protocols....357
reverse-path forwarding (RPF)....356
routing modes *See* multicast routing modes
S,G notation....355
SAP and SDP *See* SAP; SDP
session announcements....360
shortest-path tree (SPT)....356
static RP....367
See also RP
subnetwork leaves and branches....354
upstream interface....354
verifying IGMP versions....382
verifying PIM mode and interface configuration....383
verifying PIM RPF routing table....384
verifying PIM RPs....383
verifying SAP and SDP configuration....382
multicast routing modes
dense mode....356
dense mode, caution for use....355
sparse mode....356
Multicast Source Discovery Protocol....357
multiplier statement
BFD
usage guidelines....329

N

n-selectors, in IS-IS NET addresses....216
neighbors *See* adjacencies, IS-IS; BGP peers; OSPF neighbors; RIP neighbors
BGP....229
NETs (network entity titles)
n-selectors....216
parts....216
system identifier....216
network entity titles *See* NETs
network interfaces
multicast, upstream and downstream....354
verifying PIM on....383
verifying RIP message exchange....150
verifying RIP on....151
network layer reachability information *See* BGP, NLRI
network service access point (NSAP) addresses for IS-IS routers....216
networks
description....4
sample BGP confederations....322
sample BGP MED use....302
sample BGP peer session....246
sample BGP route reflector (one cluster)....305
sample BGP route reflectors (cluster of clusters)....306
sample BGP route reflectors (multiple clusters)....305
sample distance-vector routing....132
sample multiarea OSPF routing....170
sample OSPF network with stubs and NSSAs....177
sample OSPF topology....212
sample poison reverse routing....134
sample route advertisement....7
sample route aggregation....8
sample routing topology....5
sample split horizon routing....133
sample unidirectional routing....134
static routing....6
next hop
qualified, for static routes....46
next term action....427
next-hop statement
CLNS
usage guidelines....79
next-table statement
usage guidelines....24
NLRI, BGP....231
no-adaptation statement
BFD (static routes)
usage guidelines....329

Copyright © 2012, Juniper Networks, Inc.

no-indirect-next-hop statement
usage guidelines....106
no-nssa-abr statement
usage guidelines....182
no-readvertise statement
usage guidelines....53
noncontiguous address filter....459
not-so-stubby areas *See* NSSAs
configuring....182
notice icons....xviii
NSAP (network service access point) addresses for IS-IS routers....216
NSAPs (network service access points)
sample configurations....79
nssa
configuring....182
nssa statement
usage guidelines....182
NSSAs (not-so-stubby areas)
description....176

O

open messages, BGP....231
Open Shortest Path First *See* OSPF
orlonger route list match type....397
OSPF
activation....165
area border routers *See* area border routers
areas....170
See also area border routers; backbone area; NSSAs; stub areas
backbone....172
backbone area *See* backbone area
configuration overview....165
configuring, router identifier....167
controlling designated router election....168
cost *See* OSPF, metrics
designated router....166
enabling, description....167, 172, 174
IPsec authentication
OSPFv2 196
OSPFv3 196
metrics....202
multiarea network....174
route cost *See* OSPF, metrics
route preference....162
route selection....202
router identifier....167
routing algorithm....162
single-area network....172
SPF....162
tags
aggregate routes....97
topological database....160
traffic control....202
OSPF (Open Shortest Path First)
injecting OSPF routes into BGP....402
sample network topology....212
verifying host reachability....213
verifying neighbors....211
verifying RIP-enabled interfaces....210
verifying routes....211
OSPF interfaces
verifying....210
OSPF metric
configuring....204
OSPF neighbors, verifying....211
OSPF reference bandwidth
configuring....204
OSPFv2
authentication
configuring....189, 191, 193
overview....187
OSPFv3
overview....164
outgoing metric (RIP)
description....144

P

p2mp-lsp-next-hop statement
RSVP
usage guidelines....61
packets
handling packet fragments (configuration editor)....488
RIP, description....133
parentheses, in syntax descriptions....xix
partial sequence number PDU (PSNP)....218
passive routes, rejection, in static routing....53
passive statement
aggregate routes
usage guidelines....98
password
for RIPv2 authentication....149
path attributes, BGP....229, 231
path cost metrics
for RIP routes, description....144
for RIP routes, modifying....145, 146

Copyright © 2012, Juniper Networks, Inc.

path selection, IS-IS....217
path-count statement
BGP
usage guidelines....269
path-selection statement
usage guidelines....265
PDUs (protocol data units)
CSNPs....218
hello PDUs....217
LSPs....218
overview....217
PSNPs....218
peering sessions *See* BGP peers; BGP sessions
permanent routes, adding....23
PGM (Pragmatic General Multicast)....357
PIM (Protocol Independent Multicast)
dense mode....357
disabling on the network management interface....367
register messages *See* PIM register messages
RPF routing table group....378
source-specific multicast (SSM)....357
sparse mode....357
static RP router....367
supported versions....353
verifying the mode....383
verifying the RP....383
PIM register messages
filtering....370
incoming, rejecting on an RP....371
outgoing, rejecting on a designated router....374
reject policy on designated router....374
reject policy on RP router....371
ping command (stateless firewall filter)....486
explanation....486
Ping Host page, output for BGP....253
pipe (|)
bit-field logical operator....453
plus sign (+), bit-field logical operator....453
point-to-multipoint LSPs
with RSVP signaling....61
poison reverse technique....133
policers
for stateless firewall filters....481
policy *See* routing policies
policy framework....419
policy statement
aggregate routes
usage guidelines....98
policy, routing
aggregate routes....98
port number (TCP or UDP), source or destination
stateless firewall filter match conditions
IPv4 traffic....436
IPv6 traffic....446
Pragmatic General Multicast....357
preference statement
aggregate routes
usage guidelines....94
CLNS static routes
usage guidelines....79
OSPF
usage guidelines....208
static routes
usage guidelines....47
preferences
active routes....265
aggregate routes....94
for static routes....46
static routes....47
prefix-length-range match type....397
prefix-list statement
usage guidelines....459
prefix-policy statement
BGP
usage guidelines....269
Primary-level entry
secondary-level entry....364
Primary-level entry only....364
priority statement
OSPF
usage guidelines....168
propagation, suppressing....408
protocol data units *See* PDUs
Protocol Independent Multicast *See* PIM
protocols
Auto-RP....357
distance vector *See* RIP
DVMRP....357
EGPs....4
IGMP *See* IGMP
IGPs....4
IS-IS *See* IS-IS
MSDP....357
multicast *See* multicast

overview....3
PGM....357
PIM dense mode *See* PIM
PIM source-specific multicast (SSM)....357
PIM sparse mode *See* PIM
RIP *See* RIP
SAP and SDP *See* SAP; SDP
PSNP (partial sequence number PDU)....218

Q

qualified-next-hop statement
CLNS
usage guidelines....79
usage guidelines....47, 338

R

reachability
verifying for a RIP network....143, 152
verifying for OSPF network hosts....213
readvertise statement
usage guidelines....53
receive routes....24
receive statement
BGP
usage guidelines....269
static routes
usage guidelines....24
reference-bandwidth statement
usage guidelines....204
reject option to static statement
usage guidelines....24
rejecting
unauthorized PIM registration....370
reverse-path forwarding *See* RPF
RIB *See* routing table
rib statement
usage guidelines....16, 18
rib-group statement
usage guidelines....18
RIB-groups, static routes....18
RIP
demand circuits
overview....153
packets....154
RIP (Routing Information Protocol)
authentication (RIPv2 only)....148
authentication (RIPv2 only), configuring....149
basic network (configuration editor)....138
distance vector protocol....131
efficiency techniques....133
maximum hop count....132
overview....131, 137
packets....133
path cost metrics *See* path cost metrics
poison reverse technique....133
requirements....137
routing policy (configuration editor)....138
split horizon technique....133
traffic control with metrics *See* path cost metrics
traffic control with metrics, configuring....145, 146
unidirectional limitations....134
verifying host reachability 143, 152
verifying RIP message exchange 150
verifying RIP-enabled interfaces 151
RIP neighbors, verifying....151
RIPng (Routing Information Protocol next generation)
overview....135
route
aggregate statement
usage guidelines....91
route advertisements
description....6
stub areas and NSSAs, to control....176
route aggregation....7
route injection....402
route list match types....397
route manipulation actions, routing policies....394
route preference
external routes....203
internal routes....203
route redistribution....402
route reflectors *See* BGP route reflectors
BGP....306
route selection
OSPF....202
preference....203
static routes, defining....47, 338
route statement
static statement
usage guidelines....24, 28
route-flap damping....408
parameters....409
router data flow....419
router identifier
configuring....167

Copyright © 2012, Juniper Networks, Inc.

routes
aggregate *See* aggregate routes
contributing....91
static *See* static routes
routing....3
advertisements....6
aggregation....7
dynamic....6
filtering routes with policies....387
forwarding tables....5
from one source to many destinations....353
in one AS with RIP....137
IS-IS *See* IS-IS
multicast *See* multicast
neighbors *See* BGP peers; OSPF neighbors; RIP neighbors
protocol overview....3
RIP *See* RIP
RIP statistics....150
RIPng *See* RIPng
routing tables....4
static *See* static routing
See also protocols; routing policies; routing solutions
Routing Engine
handling packet fragments for (configuration editor)....488
protecting against DoS attacks....481
protecting against untrusted services and protocols (configuration editor)....476
routing information base *See* routing table
Routing Information Protocol *See* RIP
routing instance
for CLNS static routes (without IS-IS)....79
routing instances
configure....13
multiple....11
types....12
routing policies
actions....394
applying....389
configuration tasks....389, 391, 399, 402, 406, 409, 413
default actions....390
ECMP flow-based forwarding....412, 413
export statement....389
final actions....390
forwarding class with source and destination....399
grouping source and destination prefixes....399
import statement....389
injecting routes from one protocol into another....402
making BGP routes less preferable....406
match conditions....392
overview....387
PIM register messages *See* PIM register messages
policy name....389
preparation....388
prepending AS paths....406
reducing update messages with flap damping....408
RIP routing policy (configuration editor)....138
route redistribution....402
route-flap damping....408
terms....390
terms, creating....391
routing solutions
BGP confederations, for scaling problems....322
BGP route reflectors, for scaling problems....306
controlling RIP traffic with the incoming metric....145
controlling RIP traffic with the outgoing metric....146
filtering unwanted services and protocols....476
handling packet fragments (configuration editor)....488
making BGP routes less preferable....406
multicast administrative scoping....356
multicast reverse-path forwarding (RPF)....356
multicast shortest-path tree (SPT)....356
NSSAs, to control route advertisement....176
poison reverse, for traffic reduction....133
preventing multicast routing loops....356
protecting against DoS attacks....481
reducing update messages with flap damping....408
routing policies....387
split horizon, for traffic reduction....133
static route control techniques....52
stub areas, to control route advertisement....176

Copyright © 2012, Juniper Networks, Inc.

routing table
controlling static routes in....52, 53
description....4
displaying static routes in....89
RPF group, for multicast....378
sample distance-vector routing....132
updates, limitations in RIP....134
verifying for RPF....384
verifying OSPF routes....211
routing tables
creating....16
default....15
default unicast....15
export local routes....18
flow routes....15
group....18
inet.2....15
inet6.0....15
instance-name.inet.0....15
instance-name.inetflow.0....15
populating....18
RP (rendezvous point)
PIM register messages, incoming, rejecting 371
PIM register messages, outgoing, stopping 374
same reject policy on RP routers in a network....370
static....367
verifying....383
RP router *See* RP
RPF (reverse-path forwarding)
description....356
routing table group....378
verifying the routing table....384
RSVP
for point-to-multipoint LSPs....61

S

S,G notation, for multicast forwarding states....355
sample configurations
firewall filter configurations....479, 486, 492
SAP (Session Announcement Protocol)
description....357
session announcements....360
verifying....382
scoping, administrative....356
SDP (Session Discovery Protocol)
description....357
session announcements....360
verifying....382
security
MD5 authentication for RIPv2....149
password authentication for RIPv2....149
security zones
interfaces
ports....365
send statement
BGP
usage guidelines....269
service filters
overview....423
Session Announcement Protocol *See* SAP; SDP
sessions
announcements, multicast....360
Shortest Path First *See* SPF algorithm
shortest-path tree....356
show bgp neighbor command....325
explanation....326
show bgp summary command....327
explanation....327
show firewall command....479, 486, 492
show firewall filter protect-RE command....487
show firewall log command....480
show igmp interface command....382
explanation....382
show interfaces lo0 command....481
show isis adjacency brief command....222
show isis adjacency extensive command....222
explanation....223
show multicast rpf command....384
explanation....384
show ospf interface command....210
explanation....210
show ospf neighbor command....211
show ospf route command....212
explanation....212
show pim interface command....383
explanation....383
show pim rps command....383
explanation....383
show rip neighbor command....151
show rip statistics command....150
show route summary command....480, 493
explanation....480
show route terse command....89

Copyright © 2012, Juniper Networks, Inc.

show sap listen command....382
explanation....382
show isis interface brief command....221
show isis interface detail command....221
explanation....221
simple authentication
configuring
OSPFv2....189
OSPFv2....187
simple filters
overview....423
simple-password statement
usage guidelines....189
single-area network, OSPF....172
source-specific multicast....357
sparse mode *See* multicast routing modes
SPF algorithm
overview....162
split horizon technique....133
SPT (shortest-path tree)....356
ssh command....480
standard stateless firewall filters
actions
nonterminating....469
terminating....468
applying....429
configuring....424
actions....427
filter names and options....425
filter terms....425
match conditions....426
protocol families....425
overview....422
stateless firewall filters
actions....435
standard stateless firewall filters....427
actions, nonterminating
standard stateless firewall filters....469
actions, terminating
standard stateless firewall filters....468
applying to an interface (configuration editor)....481
basic use
filtering data packets....475
filtering local packets....475
handling packet fragments....488
displaying configurations....479, 486, 492
displaying statistics....487
examples
matching on IPv6 flags....493
filter names and options
standard stateless firewall filters....425
filter terms....433
standard stateless firewall filters....425
filtering router transit traffic
overview....475
filtering Routing Engine traffic
overview....475
handling packet fragments
overview....488
handling packet fragments (configuration editor)....488
match condition categories
matching on address classes....467
matching on address prefixes....459
matching on bit-field values....453
matching on numeric values....458
matching on text strings....458
match conditions....434
standard stateless firewall filters....426
overview....421
policers for....481
protecting the Routing Engine against TCP floods....481
protecting the Routing Engine against untrusted protocols (configuration editor)....476
protecting the Routing Engine against untrusted services (configuration editor)....476
protocol families....431
standard stateless firewall filters....425
sample terms, to filter fragments....488
sample terms, to filter services and protocols....476
standard stateless firewall filters....422
type
overview....431
types....423
verifying actions....480
verifying configuration....479, 486, 492
verifying flood protection....486
verifying packet logging....480
static routes....24, 28
BFD....329
CLNS VPNs (without IS-IS)....79
configuring basic routes (configuration editor)....24
configuring on logical systems....35

controlling....52
controlling in routing and forwarding tables....53
default properties....82
default properties, setting....83
defining route selection....47, 338
IPv6....28
preferences....46
preventing readvertisement....53
qualified next hops....46
rejecting passive traffic....53
route retention....53
verifying....89
static routing
description....6
overview....23
See also static routes
static RP router....367
See also RP
static statement
usage guidelines....24, 28
statistics
RIP....150
stateless firewall filters....487
stub areas
configuring....178
description....176
stub statement
usage guidelines....178
sub-ASs, BGP....321
subautonomous systems, BGP....321
subnetworks
description....4
route aggregation....8
subnetworks, multicast leaves and branches....354
summaries statement
usage guidelines....178
support, technical *See* technical support
syntax conventions....xviii
system identifier, IS-IS
all zeros not supported....216
formats, MAC or IP address....216
identifier-to-hostname mapping....216
overview....216

T

tag statement
aggregate routes
usage guidelines....97
TCP policers....481
technical support
contacting JTAC....xx
telnet command....486
terms
in a routing policy....390
in a routing policy, creating....391
through route list match type....397
to statement, routing policy match conditions....392
topology
point-to-multipoint LSPs....60
sample BGP confederations....322
sample BGP MED use....302
sample BGP peer session....246
sample BGP route reflector (one cluster)....305
sample BGP route reflectors (cluster of clusters)....306
sample BGP route reflectors (multiple clusters)....305
sample distance-vector routing....132
sample multiarea OSPF routing....170
sample OSPF network....212
sample OSPF network with stubs and NSSAs....177
sample poison reverse routing....134
sample route advertisement....7
sample route aggregation....8
sample router network....5
sample split horizon routing....133
sample static route....6
sample unidirectional routing....134
totally stubby areas
configuring....178
description....176
traceoptions statement
BFD
usage guidelines....332
routing protocols
description....498
Traceroute page
results for OSPF....213
results for RIP....143, 152
tracing operations
routing protocols....498
traffic
controlling with incoming RIP metric....145
controlling with outgoing RIP metric....146
traffic control
OSPF....202

Copyright © 2012, Juniper Networks, Inc.

traffic-engineering statement
usage guidelines.....61

U

unicast RPF
example configuration.....116
fail filters.....116
unicast-reverse-path statement
usage guidelines.....116
unnumbered SONET/SDH interfaces
as next-hop interface for static routes.....47
update messages
BGP.....231
upstream interfaces.....354
See also multicast
upto route list match type.....397

V

verification
aggregate route.....103, 114
CLNS static routes.....81
firewall filter handles fragments.....493
IGMP version.....382
martian routes.....128
multicast SAP and SDP.....382
network interfaces.....35, 43, 46, 78, 288
OSPF host reachability.....213
OSPF neighbors.....211
OSPF routes.....211
OSPF-enabled interfaces.....210
PIM mode and interface configuration.....383
PIM RP address.....383
PIM RPF routing table.....384
RIP host reachability.....143, 152
RIP message exchange.....150
RIP-enabled interfaces.....151
routing table groups.....21
stateless firewall filter actions.....480
stateless firewall filter flood protection.....486
stateless firewall filter operation.....480
stateless firewall filters.....479, 486, 492
stateless firewall statistics.....487
static
route...18, 27, 31, 39, 51, 58, 87, 122, 336, 341, 350
static routes in the routing table.....89
tracing.....501
version statement
BFD
usage guidelines.....329
virtual link, through the backbone area.....170

Copyright © 2012, Juniper Networks, Inc.