Junos® OS

Routing Protocols Overview

Modified: 2018-12-05

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS Routing Protocols Overview

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.

END USER LICENSE AGREEMENT

Table of Contents

About the Documentation . . . ix
Documentation and Release Notes . . . ix
Using the Examples in This Manual . . . ix
Merging a Full Example . . . x
Merging a Snippet . . . x
Documentation Conventions . . . xi
Documentation Feedback . . . xiii
Requesting Technical Support . . . xiii
Self-Help Online Tools and Resources . . . xiv
Opening a Case with JTAC . . . xiv

Part 1 Overview

Chapter 1 **Understanding IP Routing . . . 3**

Routing Databases Overview . . . 3
Routing Protocol Databases . . . 4
Junos OS Routing Tables . . . 4
Networks and Subnetworks . . . 5
Forwarding Tables . . . 6
How the Routing and Forwarding Tables Are Synchronized . . . 7
NetFlow V9 Support . . . 8
Routing Instances Overview . . . 8
Route Preferences Overview . . . 12
Autonomous Systems . . . 12
Alternate and Tiebreaker Preferences . . . 12
Multiple Active Routes . . . 13
Dynamic and Static Routing . . . 13
Route Advertisements . . . 14
Route Aggregation . . . 15
Understanding Route Preference Values (Administrative Distance) . . . 16
Understanding BGP Path Selection . . . 17
Routing Table Path Selection . . . 20
BGP Table path selection . . . 21
Effects of Advertising Multiple Paths to a Destination . . . 22
Equal-Cost Paths and Load Sharing Overview . . . 22
Routing Protocol Process Overview for EX Series Switches . . . 22

Chapter 2 Overview of IPv6 Routing 25
IPv6 Overview 26
IPv6 Packet Headers 26
Header Structure 27
Extension Headers 27
IPv6 Addressing 27
Address Representation 28
Address Types 28
Address Scope 28
Address Structure 29
Understanding IPv6 29
What is IPv6? 29
IPv6 Address Format 30
Implementations at Juniper Networks 31
IPv4 and IPv6 Collaboration 32
Supported IPv6 Standards 33
IPv6 Support on ACX Series Universal Metro Routers 36

Part 2 Monitoring and Troubleshooting

Chapter 3 Monitoring Networks 43
Example: Tracing Global Routing Protocol Operations 43

Chapter 4 Troubleshooting Network Issues 49
Working with Problems on Your Network 49
Isolating a Broken Network Connection 50
Identifying the Symptoms of a Broken Network Connection 51
Isolating the Causes of a Network Problem 52
Taking Appropriate Action for Resolving the Network Problem 53
Evaluating the Solution to Check Whether the Network Problem Is Resolved . . . 54

Part 3 Configuration Statements and Operational Commands

Chapter 5 Operational Commands 59
show | display rfc5952 60

List of Figures

Part 1	**Overview**
Chapter 1	**Understanding IP Routing 3**
	Figure 1: Simple Network Topology 6
	Figure 2: Synchronizing Routing Exchange Between the Routing and Forwarding Tables 8
	Figure 3: Static Routing Example 14
	Figure 4: Route Advertisement 14
	Figure 5: Route Aggregation 15
Chapter 2	**Overview of IPv6 Routing 25**
	Figure 6: IPv4 and IPv6 Header Comparison 30
Part 2	**Monitoring and Troubleshooting**
Chapter 4	**Troubleshooting Network Issues 49**
	Figure 7: Process for Diagnosing Problems in Your Network 50
	Figure 8: Network with a Problem 50

List of Tables

About the Documentation ix
Table 1: Notice Icons xi
Table 2: Text and Syntax Conventions xii

Part 1 Overview

Chapter 1 Understanding IP Routing 3
Table 3: Default Route Preference Values 16

Chapter 2 Overview of IPv6 Routing 25
Table 4: IPv6 Host Portion Techniques 31
Table 5: IPv4 and IPv6 Collaboration Strategies 32

Part 2 Monitoring and Troubleshooting

Chapter 4 Troubleshooting Network Issues 49
Table 6: Checklist for Working with Problems on Your Network 49

About the Documentation

- Documentation and Release Notes on page ix
- Using the Examples in This Manual on page ix
- Documentation Conventions on page xi
- Documentation Feedback on page xiii
- Requesting Technical Support on page xiii

Documentation and 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 https://www.juniper.net/documentation/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at https://www.juniper.net/books.

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

```
system {
  scripts {
    commit {
      file ex-script.xsl;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

```
[edit]
user@host# load merge /var/tmp/ex-script.conf
load complete
```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {
  file ex-script-snippet.xsl; }
```


2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see CLI Explorer.

Documentation Conventions

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

Table 1: Notice Icons

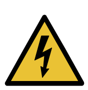
Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a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

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

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 or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User 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; 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)	Encloses 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)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies 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.	

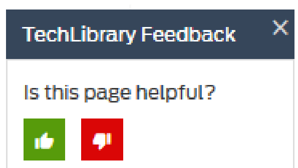
Table 2: Text and Syntax Conventions (continued)

Convention	Description	Examples
GUI Conventions		
Bold text like this	Represents 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 menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the Juniper Networks TechLibrary site, and do one of the following:



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.

- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software 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 Partner Support Service support contract, or are covered under warranty, and need post-sales 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 https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://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: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: https://www.juniper.net/cm/

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

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 https://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, see https://www.juniper.net/support/requesting-support.html.

PART 1

Overview

- Understanding IP Routing on page 3
- Overview of IPv6 Routing on page 25

CHAPTER 1

Understanding IP Routing

- Routing Databases Overview on page 3
- Routing Instances Overview on page 8
- Route Preferences Overview on page 12
- Understanding Route Preference Values (Administrative Distance) on page 16
- Understanding BGP Path Selection on page 17
- Equal-Cost Paths and Load Sharing Overview on page 22
- Routing Protocol Process Overview for EX Series Switches on page 22

Routing Databases Overview

Routing is the transmission of packets from a source to a destination address. A routing protocol determines the path by which the packets are forwarded, shares information with immediate neighbor devices and other devices in the network, and adjusts to changing network conditions.

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.

The Junos® operating system (Junos OS) maintains two databases for routing information:

- Routing table—Contains all the routing information learned by all routing protocols.
- Forwarding table—Contains the routes actually used to forward packets through the router.

In addition, the interior gateway protocols (IGPs), IS-IS, and OSPF maintain link-state databases.

This section includes the following topics:

- Routing Protocol Databases on page 4
- Junos OS Routing Tables on page 4
- Networks and Subnetworks on page 5
- Forwarding Tables on page 6

- How the Routing and Forwarding Tables Are Synchronized on page 7
- NetFlow V9 Support on page 8

Routing Protocol Databases

Each IGP routing protocol maintains a database of the routing information it has learned from other routers running the same protocol and uses this information as defined and required by the protocol. 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. IS-IS and OSPF use the routing information they received to maintain link-state databases, which they use to determine which adjacent neighbors are operational and to construct network topology maps. 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.

IS-IS and OSPF use the Dijkstra algorithm, and RIP and RIPng use the Bellman-Ford algorithm to determine the best route or routes (if there are multiple equal-cost routes) to reach each destination and install these routes into the Junos OS routing table.

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.

When you configure a protocol on an interface, you must also configure a protocol family on that interface.

Junos OS Routing Tables

The Junos OS routing table is used by the routing protocol process to maintain its database of routing information. In this table, the routing protocol process stores statically configured routes, directly connected interfaces (also called *direct routes* or *interface routes*), and all routing information learned from all routing protocols. The routing protocol process uses this collected routing information to select the *active route* to each destination, which is the route that actually is used to forward packets to that destination. 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.

By default, the Junos OS maintains three routing tables: one for unicast routes, another for multicast routes, and a third for MPLS. You can configure additional routing tables to support situations where you need to separate a particular group of routes or where you need greater flexibility in manipulating routing information. In general, most operations can be performed without resorting to the complexity of additional routing tables. However, creating additional routing tables has several specific uses, including importing interface routes into more than one routing table, applying different routing policies when exporting the same route to different peers, and providing greater flexibility with incongruent multicast topologies.

Each routing table is identified by a name, which consists of the protocol family followed by a period and a small, nonnegative integer. The protocol family can be **inet** (Internet), **iso** (ISO), or **mpls** (MPLS). The following names are reserved for the default routing tables maintained by the Junos OS:

- **inet.0**—Default IP version 4 (IPv4) unicast routing table
- **inet6.0**—Default IP version 6 (IPv6) unicast routing table
- **instance-name.inet.0**—Unicast routing table for a particular routing instance
- **inet.1**—Multicast forwarding cache
- **inet.2**—Unicast routes used for multicast reverse path forwarding (RPF) lookup
- **inet.3**—MPLS routing table for path information
- **mpls.0**—MPLS routing table for label-switched path (LSP) next hops

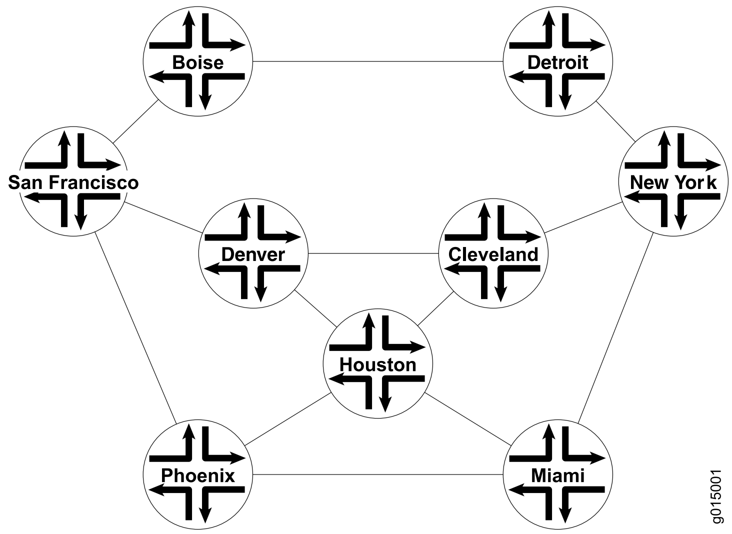
NOTE: For clarity, this topic contains general discussions of routing tables as if there were only one table. However, when it is necessary to distinguish among the routing tables, their names are explicitly used.

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.

Figure 1 on page 6 shows a simple network of routers.

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 6 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.

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.

Forwarding Tables

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.

The Junos OS installs all active routes from the routing table into the forwarding table. The active routes are used to forward packets to their destinations.

The Junos OS kernel maintains a master copy of the forwarding table. It copies the forwarding table to the Packet Forwarding Engine, which is the part of the router responsible for forwarding packets.

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.

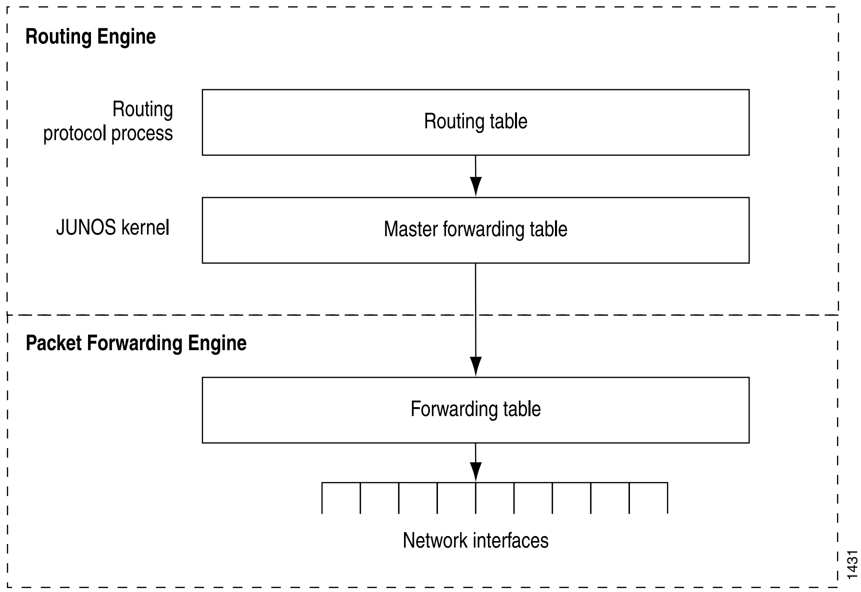
In the network shown in Figure 1 on page 6, because the path with the fewest number of hops from San Francisco to Miami is through Phoenix, the forwarding table distills all the possible San Francisco-Miami routes into the single route through Phoenix. All traffic with a destination address of Miami 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.

How the Routing and Forwarding Tables Are Synchronized

The Junos OS routing protocol process is responsible for synchronizing the routing information between the routing and forwarding tables. To do this, the routing protocol process calculates the active routes from all the routes in the routing table and installs them into the forwarding table. The routing protocol process then copies the forwarding table to the router's Packet Forwarding Engine, the part of the router that forwards packets. Figure 2 on page 8 illustrates how the routing tables are synchronized.

Figure 2: Synchronizing Routing Exchange Between the Routing and Forwarding Tables



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.

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

For more information, see *Monitoring, Sampling, and Collection Services Interfaces Feature Guide*.

Related Documentation

- *Understanding Junos OS Routing Tables*

Routing Instances Overview

You can create multiple instances of BGP, IS-IS, LDP, Multicast Source Discovery Protocol (MSDP), OSPF version 2 (usually referred to simply as OSPF), OSPF version 3 (OSPFv3), Protocol Independent Multicast (PIM), RIP, RIP next generation (RIPng), and static routes by including statements at the following hierarchy levels:

- **[edit routing-instances *routing-instance-name* protocols]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]**

Only one instance of each protocol can be configured in a single routing instance.

NOTE: You can also create multiple routing instances for separating routing tables, routing policies, and interfaces for individual DHCP wholesale subscribers (retailers) in a layer 3 wholesale network. For information about how to configure layer 3 wholesale network services, see the *Junos OS Broadband Subscriber Management and Services Library*.

A routing instance is a collection of routing tables, interfaces, and routing protocol parameters. The set of interfaces belongs to the routing tables, and the routing protocol parameters control the information in the routing tables. 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.

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**, the corresponding IP unicast table is **my-instance.inet.0**. All routes for **my-instance** are installed into **my-instance.inet.0**.

NOTE: 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.

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)

  NOTE: The commit operation fails, if the same logical interface is configured for both layer 2 circuit and ccc connection.

- Routing option configurations

You can configure 12 types of routing instances:

- Ethernet VPN (EVPN) (MX Series routers only)—Use this routing instance type to connect a group of dispersed customer sites using a Layer 2 virtual bridge.
- Forwarding—Use this routing instance type 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.
- Internet Multicast over MPLS—Use this routing instance type to provide support for ingress replication provider tunnels to carry IP multicast data between routers through an MPLS cloud, using MBGP or next-generation MVPN.

- Layer 2 Backhaul VPN—(MX Series routers only) Use this routing instance type to provide support for Layer 2 wholesale VLAN packets with no existing corresponding logical interface. When using this instance, the router learns both the outer tag and inner tag of the incoming packets, when the **instance-role** statement is defined as **access**, or the outer VLAN tag only, when the **instance-role** statement is defined as **nni**.
- Layer2-control—(MX Series routers only) Use this routing instance type for RSTP or MSTP in customer edge interfaces of a VPLS routing instance. This instance type cannot be used if the customer edge interface is multihomed to two provider edge interfaces. If the customer edge interface is multihomed to two provider edge interfaces, use the default BPDU tunneling.
- Layer 2 VPN—Use this routing instance type for Layer 2 virtual private network (VPN) implementations.
- MPLS forwarding—Use this routing instance type to provide support for protection against label spoofing or errant label injection across autonomous system border routers (ASBRs).
- Nonforwarding—Use this routing instance type 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.
- Virtual router—Similar to a VPN routing and forwarding instance type, but used for non-VPN-related applications. There are no virtual routing and forwarding (VRF) import, VRF export, VRF target, or route distinguisher requirements for this instance type.
- Virtual switch—(MX Series routers only) Use the virtual switch instance type to isolate a LAN segment with its Spanning Tree Protocol (STP) instance and separates its VLAN identifier space. For more detail information about configuring a virtual switch, see the *Junos OS Layer 2 Switching and Bridging Library* .
- VPLS—Use the virtual private local-area network service (VPLS) routing instance type for point-to-multipoint LAN implementations between a set of sites in a VPN.
- VRF—Use the VPN routing and forwarding routing (VRF) instance type for Layer 3 VPN implementations. This 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. Each VRF instance corresponds with a forwarding table. Routes on an interface go into the corresponding forwarding table.

Configure global routing options and protocols for the master instance by including statements at the **[edit protocols]** and **[edit routing-options]** hierarchy levels. Routes are installed into the master routing instance **inet.0** by default, unless a routing instance is specified.

Multiple instances of BGP, OSPF, and RIP are used for Layer 3 VPN implementation. The multiple instances of BGP, OSPF, and RIP keep routing information for different VPNs 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.

Forwarding instances are used to implement filter-based forwarding for Common Access Layer applications.

PIM instances are used to implement multicast over VPN applications.

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 policy. 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.

Layer 2 VPN instances are used for Layer 2 VPN implementation.

Virtual router instances are similar to a VPN routing and forwarding instance type, but used for non-VPN-related applications. There are no VRF import, VRF export, VRF target, or route distinguisher requirements for this instance type.

Use the VPLS routing instance type for point-to-multipoint LAN implementations between a set of sites in a VPN.

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

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 VPNs Library for Routing Devices*
- *Junos OS Layer 2 Switching and Bridging Library*
- *instance-type*

Route Preferences Overview

For unicast routes, the Junos OS routing protocol process uses the information in its routing table, along with the properties set in the configuration file, to choose an *active route* for each destination. While the Junos OS might know of many routes to a destination, the active route is the preferred route to that destination and is the one that is installed in the forwarding table and used when actually routing packets.

The routing protocol process generally determines the active route by selecting the route with the lowest preference value. The preference value is an arbitrary value in the range from 0 through 4,294,967,295 ($2^{32} – 1$) that the software uses to rank routes received from different protocols, interfaces, or remote systems.

The preference value is used to select routes to destinations in external autonomous systems (ASs) or routing domains; it has no effect on the selection of routes within an AS (that is, within an interior gateway protocol [IGP]). Routes within an AS are selected by the IGP and are based on that protocol's metric or cost value.

This section includes the following topics:

- Autonomous Systems on page 12
- Alternate and Tiebreaker Preferences on page 12
- Multiple Active Routes on page 13
- Dynamic and Static Routing on page 13
- Route Advertisements on page 14
- Route Aggregation on page 15

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.

Alternate and Tiebreaker Preferences

The Junos OS provides support for alternate and tiebreaker preferences, and some of the routing protocols, including BGP and label switching, use these additional preferences. With these protocols, you can specify a primary route preference (by including the **preference** statement in the configuration), and a secondary preference that is used as a tiebreaker (by including the **preference2** statement).

In order to use common comparison routines, Junos OS stores the 1's complement of the **LocalPref** value in the **Preference2** field. For example, if the **LocalPref** value for Route 1 is 100, the **Preference2** value is -101. If the **LocalPref** value for Route 2 is 155, the **Preference2** value is -156. Route 2 is preferred because it has a higher **LocalPref** value and a lower **Preference2** value.

You can also mark route preferences with additional route tiebreaker information by specifying a color and a tiebreaker color (by including the **color** and the tiebreaker **color2** statements in the configuration). **color** and **color2** statements are even finer-grained preference values that Junos OS uses when **preference** and **preference2** statements fail to break the tie during route selection.

The software uses a 4-byte value to represent the route preference value. When using the preference value to select an active route, the software first compares the primary route preference values, choosing the route with the lowest value. If there is a tie and a secondary preference has been configured, the software compares the secondary preference values, choosing the route with the lowest value. The secondary preference values must be included in a set for the preference values to be considered.

Multiple Active Routes

The IGPs compute equal-cost multipath next hops, and IBGP picks up these next hops. When there are multiple, equal-cost next hops associated with a route, the routing protocol process installs only one of the next hops in the forwarding path with each route, randomly selecting which next hop to install. For example, if there are 3 equal-cost paths to an exit routing device and 900 routes leaving through that routing device, each path ends up with about 300 routes pointing at it. This mechanism provides load distribution among the paths while maintaining packet ordering per destination.

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.

Random selection of equal-cost multipath occurs independently for **inet.0** and **inet.3** tables. This can lead to a single prefix showing different bestpaths for **inet.0** vs **inet.3**.

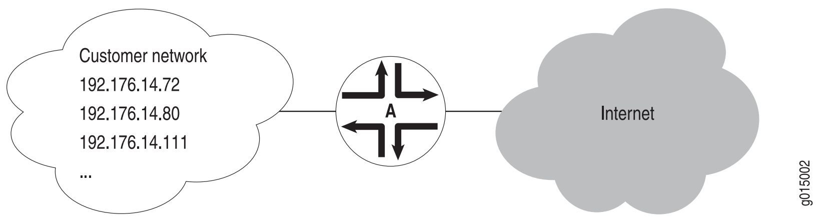
Dynamic and Static Routing

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 3 on page 14 shows a network that uses static routes.

Figure 3: Static Routing Example



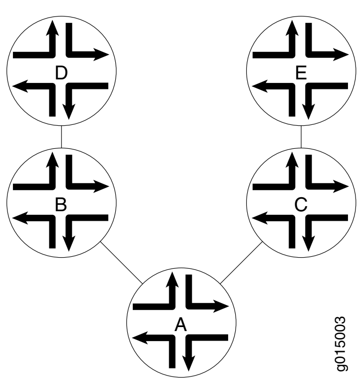
In Figure 3 on page 14, the customer routes in the **192.176.14/24** subnetwork are static routes. These are hard links to specific customer hosts that never change. Because all traffic destined for any of these routes is forwarded through Router A, these routes are included as static routes in Router 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 4 on page 14.

Figure 4: Route Advertisement



In Figure 4 on page 14, 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 5 on page 15.

Figure 5: Route Aggregation

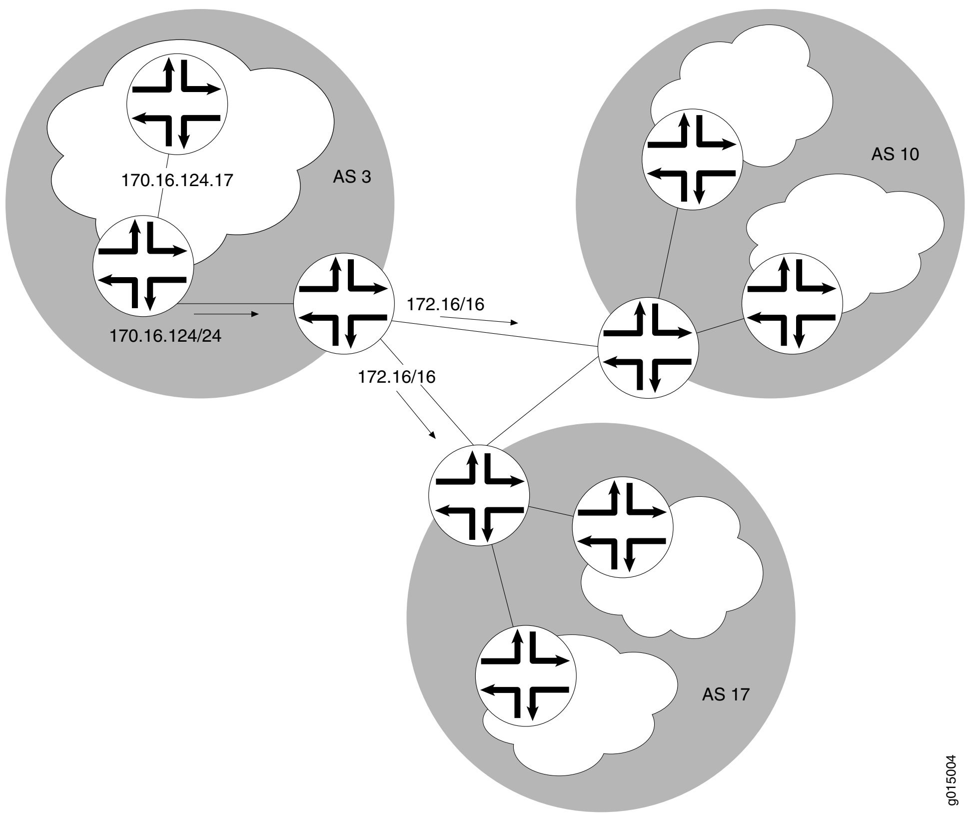


Figure 5 on page 15 shows three different ASs. Each AS contains multiple subnetworks with thousands of host addresses. To allow traffic to be sent from any host to any host, the routing tables for each host must include a route for each destination. For the routing tables to include every combination 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.

Understanding Route Preference Values (Administrative Distance)

The Junos OS routing protocol process assigns a default preference value (also known as an *administrative distance*) to each route that the routing table receives. The default value depends on the source of the route. The preference value is a value from 0 through 4,294,967,295 (2^{32} – 1), with a lower value indicating a more preferred route. Table 3 on page 16 lists the default preference values.

Table 3: Default Route Preference Values

How Route Is Learned	Default Preference	Statement to Modify Default Preference
Directly connected network	0	–
System routes	4	–
Static and Static LSPs	5	*static*
Static LSPs	6	–
RSVP-signaled LSPs	7	RSVP **preference** as described in the *MPLS Applications Feature Guide*
LDP-signaled LSPs	9	LDP **preference**, as described in the *MPLS Applications Feature Guide*
OSPF internal route	10	OSPF *preference*
access-internal route	12	–
access route	13	–
IS-IS Level 1 internal route	15	IS-IS *preference*
IS-IS Level 2 internal route	18	IS-IS *preference*

Table 3: Default Route Preference Values (continued)

How Route Is Learned	Default Preference	Statement to Modify Default Preference
Redirects	30	–
Kernel	40	–
SNMP	50	–
Router discovery	55	–
RIP	100	RIP *preference*
RIPng	100	RIPng *preference*
PIM	105	*Multicast Protocols Feature Guide*
DVMRP	110	*Multicast Protocols Feature Guide*
Aggregate	130	*aggregate*
OSPF AS external routes	150	OSPF *external-preference*
IS-IS Level 1 external route	160	IS-IS *external-preference*
IS-IS Level 2 external route	165	IS-IS *external-preference*
BGP	170	BGP *preference*, *export*, *import*
MSDP	175	*Multicast Protocols Feature Guide*

In general, the narrower the scope of the statement, the higher precedence its preference value is given, but the smaller the set of routes it affects. To modify the default preference value for routes learned by routing protocols, you generally apply routing policy when configuring the individual routing protocols. You also can modify some preferences with other configuration statements, which are indicated in the table.

Related Documentation

- *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide*

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.

By default, only the 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.

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.

    BGP compares the type of IGP metric before comparing the metric value itself in rt_metric2_cmp. For example, BGP routes that are resolved through IGP are preferred over discarded or rejected next-hops that are of type RTM_TYPE_UNREACH. Such routes are declared inactive because of their metric-type.

11. If both paths are external, prefer the currently active path to minimize route-flapping. This rule is not used if any one of the following conditions is true:

    - **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 a primary route over a secondary route. A primary route is one that belongs to the routing table. A secondary route is one that is added to the routing table through an export policy.

13. 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.

14. Prefer the path with the shortest cluster list length. The length is 0 for no list.

15. Prefer the path from the peer with the lowest peer IP address.

Routing Table Path Selection

The shortest AS path 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 step of the algorithm by including the **as-path-ignore** option.

NOTE: **Starting with Junos OS Release 14.1R8, 14.2R7, 15.1R4, 15.1F6, and 16.1R1, the as-path-ignore option is supported for routing instances.**

The routing process path selection takes place before BGP hands off the path to the routing table to makes its decision. 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:

- Emulate the Cisco IOS default behavior (**cisco-non-deterministic**). This mode evaluates routes in the order that they are received and does not group them according to their neighboring AS. With **cisco-non-deterministic** mode, the active path is always first. All inactive, 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.

  As an example, suppose you have three path advertisements for the 192.168.1.0 /24 route:

  - Path 1—learned through EBGP; AS Path of 65010; MED of 200
  - Path 2—learned through IBGP; AS Path of 65020; MED of 150; IGP cost of 5
  - Path 3—learned through IBGP; AS Path of 65010; MED of 100; IGP cost of 10

  These advertisements are received in quick succession, within a second, in the order listed. Path 3 is received most recently, so the routing device compares it against path 2, the next most recent advertisement. The cost to the IBGP peer is better for path 2, so the routing device eliminates path 3 from contention. When comparing paths 1 and

2, the routing device prefers path 1 because it is received from an EBGP peer. This allows the routing device to install path 1 as the active path for the route.

NOTE: **We do not recommend using this configuration option in your network. It is provided solely for interoperability to allow all routing devices in the network to make consistent route selections.**

- Always comparing MEDs whether or not the peer ASs of the compared routes are the same (**always-compare-med**).
- Override the rule that If both paths are external, the currently active path is preferred (**external-router-id**). Continue with the next step (Step 12) in the path-selection process.
- Adding the IGP cost to the next-hop destination to the MED value before comparing MED values for path selection (**med-plus-igp**).

  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.

BGP Table path selection

The following parameters are followed for BGP's path selection:

1. Prefer the highest local-preference value.
2. Prefer the shortest AS-path length.
3. Prefer the lowest origin value.
4. Prefer the lowest MED value.
5. Prefer routes learned from an EBGP peer over an IBGP peer.
6. Prefer best exit from AS.
7. For EBGP-received routes, prefer the current active route.
8. Prefer routes from the peer with the lowest Router ID.
9. Prefer paths with the shortest cluster length.
10. Prefer routes from the peer with the lowest peer IP address. Steps 2, 6 and 12 are the RPD criteria.

Effects of Advertising Multiple Paths to a Destination

BGP advertises only the active path, unless you configure BGP to advertise multiple paths to a destination.

Suppose a routing device has in its routing table four paths to a destination and is configured to advertise up to three paths (**add-path send path-count 3**). The three paths are chosen based on path selection criteria. That is, the three best paths are chosen in path-selection order. The best path is the active path. This path is removed from consideration and a new best path is chosen. This process is repeated until the specified number of paths is reached.

Release History Table

Release	Description
14.1R8	Starting with Junos OS Release 14.1R8, 14.2R7, 15.1R4, 15.1F6, and 16.1R1, the **as-path-ignore** option is supported for routing instances.

Related Documentation

- *Example: Ignoring the AS Path Attribute When Selecting the Best Path*
- *Examples: Configuring BGP MED*
- *Example: Advertising Multiple BGP Paths to a Destination*

Equal-Cost Paths and Load Sharing Overview

For equal-cost paths, load sharing is based on the BGP next hop. For example, if four prefixes all point to a next hop and there is more than one equal-cost path to that next hop, the routing protocol process uses a hash algorithm to choose the path among the four prefixes. Also, for each prefix, the routing protocol process installs a single forwarding entry pointing along one of the paths. The routing software does not rehash the path taken as prefixes pointing to the next hop come and go, but it does rehash if the number of paths to the next hop changes. Because a prefix is tied to a particular path, packet reordering should not happen. The degree of load sharing improves as the number of prefixes increases.

Routing Protocol Process Overview for EX Series Switches

Junos OS is based on the FreeBSD Unix operating system. The open source software is modified and hardened to operate in the device's specialized environment. For example, some executables have been deleted, while other utilities were de-emphasized. Additionally, certain software processes were added to enhance the routing functionality. The result of this transformation is the kernel, the heart of the Junos OS software.

The kernel is responsible for operating multiple processes that perform the actual functions of the device. Each process operates in its own protected memory space, while the communication among all the processes is still controlled by the kernel. This separation provides isolation between the processes, and resiliency in the event of a process failure. This is important in a core routing platform because a single process failure does not cause the entire device to cease functioning.

Some of the common software processes include the routing protocol process (rpd) that controls the device's protocols, the device control process (dcd) that controls the device's interfaces, the management process (mgd) that controls user access to the device, the chassis process (chassisd) that controls the device's properties itself, and the Packet Forwarding Engine process (pfed) that controls the communication between the device's Packet Forwarding Engine and the Routing Engine. The kernel also generates specialized processes as needed for additional functionality, such as SNMP, the Virtual Router Redundancy Protocol (VRRP), and Class of Service (CoS).

The routing protocol process is a software process within the Routing Engine software, which controls the routing protocols that run on the device. Its functionality includes all protocol messages, routing table updates, and implementation of routing policies.

The routing protocol process starts all configured routing protocols and handles all routing messages. It maintains one or more routing tables, which consolidate the routing information learned from all routing protocols. From this routing information, the routing protocol process determines the active routes to network destinations and installs these routes into the Routing Engine's forwarding table. Finally, it implements routing policy, which allows you to control the routing information that is transferred between the routing protocols and the routing table. Using routing policy, you can filter and limit the transfer of information as well as set properties associated with specific routes.

Related Documentation

- *show system processes*
- *show task*
- *show task memory*

CHAPTER 2

Overview of IPv6 Routing

- IPv6 Overview on page 26
- Understanding IPv6 on page 29
- Supported IPv6 Standards on page 33
- IPv6 Support on ACX Series Universal Metro Routers on page 36

IPv6 Overview

IP version 6 (IPv6) is the latest version of IP. IP enables numerous nodes on different networks to interoperate seamlessly. IP version 4 (IPv4) is currently used in intranets and private networks, as well as the Internet. IPv6 is the successor to IPv4, and is based for the most part on IPv4.

IPv4 has been widely deployed and used to network the Internet today. With the rapid growth of the Internet, enhancements to IPv4 are needed to support the influx of new subscribers, Internet-enabled devices, and applications. IPv6 is designed to enable the global expansion of the Internet.

IPv6 builds upon the functionality of IPv4, providing improvements to addressing, configuration and maintenance, and security.

IPv6 offers the following benefits:

- Expanded addressing capabilities—IPv6 provides a larger address space. IPv6 addresses consist of 128 bits, while IPv4 addresses consist of 32 bits. 128-bit addressing increases the address space by approximately 10^29 unique addresses, enough to last for the forseeable future.
- Header format simplification—IPv6 packet header format is designed to be efficient. IPv6 standardizes the size of the packet header to 40 bytes, divided into 8 fields.
- Improved support for extensions and options—Extension headers carry Internet-layer information and have a standard size and structure.
- Flow labeling capability—Flow labels provide consistent handling of packets belonging to the same flow.
- Improved privacy and security—IPv6 supports extensions for authentication and data integrity, which enhance privacy and security.

This section discusses the following topics:

- IPv6 Packet Headers on page 26
- IPv6 Addressing on page 27

IPv6 Packet Headers

IPv6 headers are different from IPv4 headers.

This section discusses the following topics that provide background information about IPv6 headers:

- Header Structure on page 27
- Extension Headers on page 27

Header Structure

IPv6 packet headers contain many of the fields found in IPv4 packet headers; some of these fields have been modified from IPv4. The 40-byte IPv6 header consists of the following 8 fields:

- Traffic class—Class-of-service (CoS) priority of the packet. Previously the type-of-service (ToS) field in IPv4. However, the semantics of this field (for example, DiffServ code points) are identical to IPv4.
- Destination address—Final destination node address for the packet.
- Flow label—Packet flows requiring a specific class of service. The flow label identifies all packets belonging to a specific flow, and routers can identify these packets and handle them in a similar fashion.
- Hop limit—Maximum number of hops allowed. Previously the time-to-live (TTL) field in IPv4.
- Next header—Next extension header to examine. Previously the protocol field in IPv4.
- Payload length—Length of the IPv6 payload. Previously the total length field in IPv4.
- Source address—Address of the source node sending the packet.
- Version—Version of IP.

Extension Headers

In IPv6, *extension headers* are used to encode optional Internet-layer information.

Extension headers are placed between the IPv6 header and the upper layer header in a packet.

Extension headers are chained together using the next header field in the IPv6 header. The next header field indicates to the router which extension header to expect next. If there are no more extension headers, the next header field indicates the upper layer header (TCP header, User Datagram Protocol [UDP] header, ICMPv6 header, an encapsulated IP packet, or other items).

IPv6 Addressing

IPv6 uses a 128-bit addressing model. This creates a much larger address space than IPv4 addresses, which are made up of 32 bits. IPv6 addresses also contain a scope field that categorizes what types of applications are suitable for the address. IPv6 does not support broadcast addresses, but instead uses multicast addresses to serve this role. In addition, IPv6 also defines a new type of address called *anycast*.

NOTE: You cannot configure a subnet zero IPv6 address because RFC 2461 reserves the subnet-zero address for anycast addresses, and Junos OS complies with the RFC.

This section discusses the following topics that provide background information about IPv6 addressing:

- Address Representation on page 28
- Address Types on page 28
- Address Scope on page 28
- Address Structure on page 29

Address Representation

IPv6 addresses consist of 8 groups of 16-bit hexadecimal values separated by colons (:). The IPv6 address format is as follows:

```
aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa
```

aaaa is a 16-bit hexadecimal value, and ***a*** is a 4-bit hexadecimal value. Following is an example of an actual IPv6 address:

```
3FFE:0000:0000:0001:0200:F8FF:FE75:50DF
```

You can omit the leading zeros, as shown:

```
3FFE:0:0:1:200:F8FF:FE75:50DF
```

You can compress 16-bit groups of zeros to the notation :: (two colons), as shown here, but only once per address:

```
3FFE::1:200:F8FF:FE75:50DF
```

Address Types

There are three types of IPv6 addresses:

- Unicast—For a single interface.
- Multicast—For a set of interfaces on the same physical medium. A packet is sent to all of the interfaces associated with the address.
- Anycast—For a set of interfaces on different physical mediums. A packet is sent to only one of the interfaces associated with this address, not to all the interfaces.

Address Scope

IPv6 addresses have *scope*, which identifies the application suitable for the address. Unicast and multicast addresses support scoping.

Unicast addresses support two types of scope: *global* scope and *local* scope. There are two types of local scope: *link-local* addresses and *site-local* addresses. Link-local unicast addresses are used within a single network link. The first ten bits of the prefix identify the address as a link-local address. Link-local addresses cannot be used outside a network link. Site-local unicast addresses are used within a site or intranet. A site consists of multiple network links, and site-local addresses identify nodes inside the intranet. Site-local addresses cannot be used outside the site.

Multicast addresses support 16 different types of scope, including node, link, site, organization, and global scope. A 4-bit field in the prefix identifies the scope.

Address Structure

Unicast addresses identify a single interface. The address consists of *n* bits for the prefix, and 128 – *n* bits for the interface ID.

Multicast addresses identify a set of interfaces. The address is made up of the first 8 bits of all ones, a 4-bit flags field, a 4-bit scope field, and a 112-bit group ID:

```
11111111 | flags | scope | group ID
```

The first octet of ones identifies the address as a multicast address. The flags field identifies whether the multicast address is a well-known address or a transient multicast address. The scope field identifies the scope of the multicast address. The 112-bit group ID identifies the multicast group.

Similar to multicast addresses, anycast addresses identify a set of interfaces. However, packets are sent to only one of the interfaces, not to all interfaces. Anycast addresses are allocated from the normal unicast address space and cannot be distinguished from a unicast address in format.

Understanding IPv6

Service providers and some enterprises are faced with growing their networks using IPv6, while continuing to serve IPv4 customers.

Juniper Networks has made significant investments in technologies and solutions that enable enterprises and service providers to meet mixed IP addressing needs even as they build out IPv6 networks as rapidly as markets and services require.

Increasingly, the public side of network address translation (NAT) devices is IPv6 rather than IPv4. Service providers cannot continue giving customers globally routable IPv4 addresses, they cannot get new globally routable IPv4 addresses for expanding their own networks, and yet they must continue to serve both IPv4 customers and new customers, all of whom are primarily trying to reach IPv4 destinations.

IPv4 and IPv6 must coexist for some number of years, and their coexistence must be transparent to end users. If an IPv4-to-IPv6 transition is successful, the end users should not even notice it.

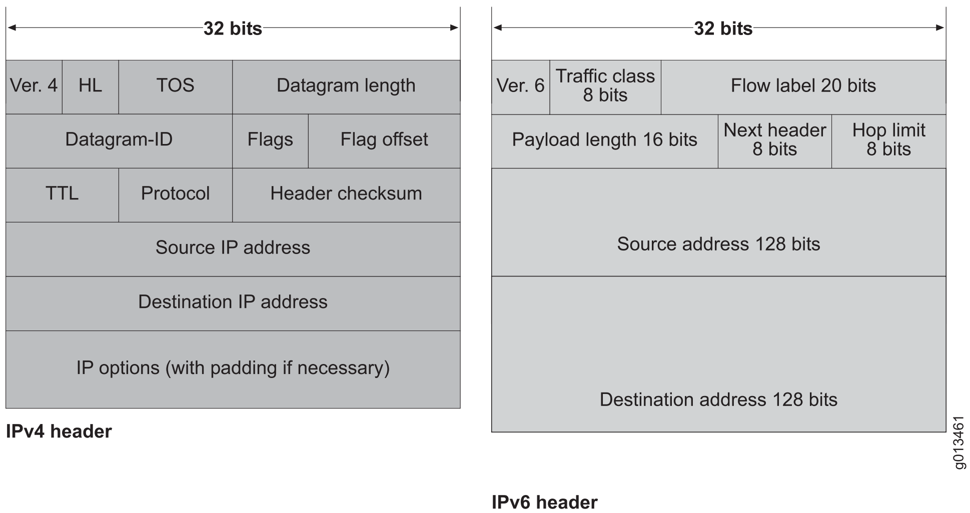
What is IPv6?

IP version 6 (IPv6) is the latest version of IP. IPv6 builds upon the functionality of IPv4, providing improvements to addressing, configuration and maintenance, and security. Juniper Networks is focused on helping service provider and enterprise customers deploy IPv6 in ways that improve current networks.

IPv6 offers the following benefits:

- Expanded addressing capabilities—IPv4 uses 32-bit addresses and can support 4.3 billion devices connected directly to the Internet. IPv6, on the other hand, uses 128-bit addresses and supports a virtually unlimited number of devices—2 to the 128th power.
- Header format simplification—IPv6 packet header format is designed to be efficient. IPv6 standardizes the size of the packet header to 40 bytes, divided into 8 fields. Figure 6 on page 30 provides a comparison between the packet headers of the two protocol versions.

Figure 6: IPv4 and IPv6 Header Comparison



- Improved support for extensions and options—Extension headers carry Internet-layer information and have a standard size and structure.
- Flow labeling capability—Flow labels provide consistent handling of packets belonging to the same flow.
- Improved privacy and security—IPv6 supports extensions for authentication and data integrity, which enhance privacy and security.

IPv6 Address Format

IPv6 addresses consist of eight hexadecimal groups. Each hexadecimal group, separated by a colon (:), consists of a 16-bit hexadecimal value. The following is an example of the IPv6 format:

xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

A group of *xxxx* represents the 16-bit hexadecimal value. Each individual *x* represents a 4-bit hexadecimal value. The following is an example of a possible IPv6 address:

4FDE:0000:0000:0002:0022:F376:FF3B:AB3F

The first sixty four bits (4FDE:0000:0000:0002) are network bits, the remaining ones are the host's interface identifier (host bits). The network portion is provided by an ISP or by the registry (ARIN or RIPE).

The length of the prefix depends on the size of your organiztion:

- Registries are assigned /23.
- ISPs are assigned /32.
- Sites are assigned /48.

Say, you are the organization that receives a /48 prefix like this: 4FDE:0000:0000:*0000*:0000:0000:0000:0000/48. This gives you two bytes (shown in itallics) in the network portion to create different networks (itallic portion: 2^16=65536 different numbers). As a shortcut, this network address space can be represented as 4FDE::/48.

To create the host portion of IPv6 address, if DHCP is not used, you have several options.

Table 4 on page 31 lists the host addressing strategies.

Table 4: IPv6 Host Portion Techniques

Ways to Create the Host Portion of an IPv6 Address	Example
Embed an IPv4 address in an IPv6 address	4FDE::101.45.75.219
Manually	4FDE::1
EUI-64	Automatically create the host portion of IPv6 address based on the MAC address of the first Ethernet interface

For an example of manually assigned host addresses, see *Example: Configuring IPv6 Static Routes*. For an example of EUI-64 assigned host addresses, see *Example: Configuring a Basic RIPng Network*.

Implementations at Juniper Networks

When deploying IPv6, you can gain a great advantage by using Juniper Networks high-end routers because IPv6 has been implemented directly in the ASICs (Application-Specific Integrated Circuit). Having IPv6 compatibility in the hardware means that IPv6 packets can be forwarded at line rate – unlike many competing routers.

After over a decade of development, the IPv6 functionality in Juniper Networks products is extensive. Junos OS, for over ten years has had IPv6 support. Juniper has a tremendous presence on various technical bodies that have specified IPv6. Juniper had already enabled IPv6 across all of its platforms and interfaces back in 2002. Juniper was at the forefront of shipping IPv6-ready firewall and VPN gear in 2004. And Juniper was the first to have its routers certified as IPv6 capable by the U.S. Defense Department in 2007.

Just to highlight a few, Junos OS fully supports the following IPv6 RFCs:

- RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*
- RFC 3513, *Internet Protocol Version 6 (IPv6) Addressing Architecture*
- RFC 2893, *Transition Mechanisms for IPv6 Hosts and Routers*

For a complete list of supported IPv6 RFCs, see "Supported IPv6 Standards" on page 33.

IPv4 and IPv6 Collaboration

IPv6 is the biggest upgrade in the 40-year history of the Internet. Forward-looking carriers and enterprises are deploying IPv6 because the Internet has run out of allocatable IP addresses using the current IPv4 standard. Juniper is putting its energy into supporting native IPv6 as well as dual-stack configurations where IPv6 runs alongside IPv4 in all of its products. Juniper fully supports an IPv4-to-IPv6 transition mechanism known as Dual-Stack Lite, and it has been a leader in another approach called 6PE for use with multiprotocol label switching (MPLS) networks.

Keep in mind that if you are going to dual stack all of your network devices, the interfaces need both an IPv6 and an IPv4 address. This raises the issue that the Internet has run out of IPv4 addresses, which is the main reason we need IPv6 in the first place. If you do not have an abundant supply of IPv4 addresses to apply to your devices, you can still use dual stacking, but you will need to conserve your supply of IPv4 addresses by using network address translation (NAT).

Building dual stacked networks with a mix of global IPv6 addresses and NAT-ed IPv4 addresses is quite feasible. Some specific solutions include carrier-grade NAT (CGN), NAT444, NAT464, and dual-stack lite.

Table 5 on page 32 lists the types of IP transition strategies supported by Juniper Networks.

Table 5: IPv4 and IPv6 Collaboration Strategies

IPv4 and IPv6 Collaboration Strategy	Purpose
Carrier-grade NAT—Sharing IPv4 addresses	To maintain IPv4 subscriber growth after IPv4 exhaustion, the remaining IPv4 addresses will have to be shared among end users. This is done with carrier-grade NAT (CGN). Rather than assigning public addresses directly to individual users, CGN "pulls back" these addresses to a more centralized Network Address Translation (NAT) point, allowing the sharing of a single public address among a much larger number of end devices. There are several variations in the deployment architecture of CGN. Dual Stack Lite (DS-Lite) and NAT44(4) are the most important ones for coexistence strategies. They are similar in the way that they enable providers to share a small set of IPv4 addresses among a large number of users. They differ in the way that packets are carried to the CGN. With DS-Lite, they are carried as IPv4 through an IPv6 tunnel; with NAT44(4) they are carried over IPv4.
NAT44(4)	NAT44(4) is an architecture that uses the NAT44 protocol to extend the life of a customer's IPv4 address pool by allowing multiple subscribers or end users to share a single public IPv4 address. NAT44(4) requires no change to the service provider's existing network infrastructure, and can be used in conjunction with 6rd for further benefits. In NAT44(4), the subscribers have their own private IPv4 (RFC1918) address space behind their customer premises equipment (CPE). The service provider translates the subscriber's address to another IPv4 address in the access network to allow better utilization of the existing public IPv4 address space by aggregating subscribers in a public IPv4 pool on the carrier-grade NAT (CGN) router.

Table 5: IPv4 and IPv6 Collaboration Strategies (continued)

IPv4 and IPv6 Collaboration Strategy	Purpose
Dual Stack Lite (DS-Lite)	DS-Lite uses tunneling and NAT44 to mitigate IPv4 address depletion while incrementally adopting IPv6. When a device in the customer network sends an IPv4 packet to any destination, the IPv4 packet is encapsulated in an IPv6 packet for transport into the provider network. The address family transition router (AFTR) decapsulates the packet back to IPv4, and uses NAT44 to translates the private IPv4 address to a public IPv4 address and delivers the packet to the Internet.
Additional Juniper Networks supported IPv4/IPv6 technologies	• NAT64—provides IPv6 to IPv4 translation allowing IPv6-only hosts to access IPv4-only hosts. • 6to4—connects IPv6 hosts or networks across an IPv4 infrastructure or Internet. • 6rd—provides rapid deployment of IPv6 service to end users over an existing IPv4 infrastructure. • IPv4/IPv6 dual stack—Junos OS supports IPv4/IPv6 dual stack, allowing concurrent independent operation of both protocols on a single router.

Related Documentation

- *Ethernet Interfaces Feature Guide for Routing Devices*
- *IPv6 Neighbor Discovery Feature Guide*
- *Class of Service Feature Guide for Routing Devices and EX9200 Switches*
- *RIP Feature Guide*
- *Examples: Configuring MLD*
- *Day One: Exploring IPv6*
- *Day One: Advanced IPv6 Configuration*
- https://www.juniper.net/ipv6

Supported IPv6 Standards

Junos OS substantially supports the following RFCs and Internet drafts, which define standards for IP version 6 (IPv6):

- RFC 1981, *Path MTU Discovery for IP version 6*
- RFC 2373, *IP Version 6 Addressing Architecture*
- RFC 2375, *Multicast Address Assignments*
- RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*
- RFC 2461, *Neighbor Discovery for IP Version 6 (IPv6)*
- RFC 2462, *IPv6 Stateless Address Autoconfiguration*
- RFC 2463, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
- RFC 2464, *Transmission of IPv6 Packets over Ethernet Networks*

- RFC 2465, *Management Information Base for IP Version 6: Textual Conventions and General Group*

  IP version 6 (IPv6) and Internet Control Message Protocol version 6 (ICMPv6) statistics are not supported.

- RFC 2472, *IP Version 6 over PPP*
- RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
- RFC 2491, *IPv6 Over Non-Broadcast Multiple Access (NBMA) networks*
- RFC 2492, *IPv6 over ATM Networks*
- RFC 2526, *Reserved IPv6 Subnet Anycast Addresses*
- RFC 2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*
- RFC 2578, *Structure of Management Information Version 2 (SMIv2)*
- RFC 2675, *IPv6 Jumbograms*
- RFC 2711, *IPv6 Router Alert Option*
- RFC 2767, *Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)*
- RFC 2784, *Generic Routing Encapsulation*
- RFC 2878, *PPP Bridging Control Protocol (BCP)*
- RFC 3306, *Unicast-Prefix-based IPv6 Multicast Addresses*
- RFC 3307, *Allocation Guidelines for IPv6 Multicast Addresses*
- RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

  Address assignment is supported with IP version 4 (IPv4) but not IP version 6 (IPv6).

- RFC 3484, *Default Address Selection for Internet Protocol version 6 (IPv6)*
- RFC 3513, *Internet Protocol Version 6 (IPv6) Addressing Architecture*
- RFC 3515, *The Session Initiation Protocol (SIP) Refer Method*
- RFC 3590, *Source Address Selection for the Multicast Listener D* (Supported for SSM include mode only)
- RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*
- RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*
- RFC 3971, *Secure Neighbor Discovery for IPv6* (No support for certification paths, anchored on trusted parties)
- RFC 3972, *Cryptographically Generated Addresses*
- RFC 4087, *IP Tunnel MIB*
- RFC 4213, *Basic Transition Mechanisms for IPv6 Hosts and Routers*

  RFC 4213 supersedes RFC 2893, *Transition Mechanisms for IPv6 Hosts and Routers*.

NOTE: On EX Series switches, except for the EX9200 Series, only dual IP layer is supported. On EX9200 Series switches, both dual IP layer and configured tunneling of IPv6 over IPv4 are supported.

- RFC 4291, *IP Version 6 Addressing Architecture*
- RFC 4292, *IP Forwarding Table MIB*
- RFC 4293, *Management Information Base for the Internet Protocol (IP)*
- RFC 4294, *IPv6 Node Requirements* (Partial support)
- RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
- RFC 4552, *Authentication/Confidentiality for OSPFv3*
- RFC 4604, *Using Internet Group Management Protocol Version 3 (IGMPv3)*
- RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*
- RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*

  Option 4b (eBGP redistribution of labeled IPv6 routes from AS to neighboring AS) is not supported.
- RFC 4861 *Neighbor Discovery for IP Version 6 (IPv6)*
- RFC 4862, *IPv6 Stateless Address Autoconfiguration*
- RFC 4890, *Recommendations for Filtering ICMPv6 Messages in Firewalls*
- RFC 4942, *IPv6 Transition/Coexistence Security Considerations*
- RFC 5072, *IP Version 6 over PPP*
- RFC 5095, *Deprecation of Type 0 Routing Headers in IPv6*
- RFC 5308, *Routing IPv6 with IS-IS*
- RFC 5340, *OSPF for IPv6* (RFC 2740 is obsoleted by RFC 5340)
- RFC 5575, *Dissemination of Flow Specification Rules*
- RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6*
- RFC 5905, *Network Time Protocol Version 4 (for IPv6)*
- RFC 5952, *A Recommendation for IPv6 Address Text Representation*
- RFC 6164, *Using 127-Bit IPv6 Prefixes on Inter-Router Links*
- RFC 6527, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol Version 3 (VRRPv3)*

  The following features are not supported:

  - Row creation
  - **Set** operation

  - **vrrpv3StatisticsPacketLengthErrors** MIB object
  - **vrrpv3StatisticsRowDiscontinuityTime** MIB object

- RFC 6583, *Operational Neighbor Discovery Problems*

  Only Prioritize NDP Activities, Tuning of the NDP Queue Rate Limit, and Queue Tuning are supported.

- Internet draft draft-ietf-l3vpn-bgp-ipv6-07.txt, *BGP-MPLS IP VPN extension for IPv6 VPN*
- Internet draft draft-ietf-idr-flow-spec-00.txt, *Dissemination of flow specification rules*
- Internet draft draft-ietf-softwire-dual-stack-lite-04.txt, *Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion*
- Internet draft draft-kato-bgp-ipv6-link-local-00.txt, *BGP4+ Peering Using IPv6 Link-local Address*

The following RFCs and Internet draft do not define standards, but provide information about IPv6 and related technologies. The IETF classifies them variously as “Experimental” or “Informational.”

- RFC 1901, *Introduction to Community-based SNMPv2*
- RFC 2767, *Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)*
- RFC 3587, *IPv6 Global Unicast Address Format*
- Internet draft draft-ietf-ngtrans-bgp-tunnel-04.txt, *Connecting IPv6 Islands across IPv4 Clouds with BGP*

  Only MP-BGP over IP version 4 (IPv4) approach is supported.

Related Documentation

- *Supported IPv4, TCP, and UDP Standards*
- *Accessing Standards Documents on the Internet*

IPv6 Support on ACX Series Universal Metro Routers

IPv6 builds upon the functionality of IPv4, providing improvements to addressing, configuration and maintenance, and security. The following IPv6 features are supported on ACX Series routers:

- **IPv6 path maximum transmission unit (MTU) discovery**

  Path MTU Discovery is used by single-source devices to determine the correct size of fragments. Path MTU Discovery is enabled for IPv6 packets by default.

- **Dynamic routes distribution through IS-IS and OSPF for IPv6**

  Routers learn routes through different routing protocols such as OSPF, BGP, or IS-IS. Learned routes are put in the routing table to enable IPv6 traffic forwarding.

- **Dual stacking (IPv4 and IPv6)**

Dual stacking allows a device to run both IPv4 and IPv6 at the same time. End nodes, routers, and switches run both protocols and use IPv6 as the preferred protocol.

- **IPv6 forwarding**

  The ACX Series port forwarding engine software supports unicast IPv6 routes and next hops. This includes basic route infrastructure, next-hop support, network infrastructure, and exception packet processing.

- **IPv6 over MPLS (6PE)**

  ACX Series Universal Metro Routers can interconnect IPv6 islands over an MPLS-enabled IPv4 network. IPv6 information is sent over the MPLS core using MG-BGP with IPv4. The BGP Next Hop field conveys the IPv4 address of the router so that MPLS LSPs can be used without explicit tunnel configuration.

- **Neighbor Discovery**

  The Neighbor Discovery protocol facilitates a substantial number of functions related to local network connectivity, datagram routing, and configuration. Both regular hosts and routers in an IPv6 environment count on the Neighbor Discovery protocol to facilitate the important exchanges of information that are necessary for proper internetwork operations. Neighbor Discovery is a messaging protocol similar to ICMP. The following functions are performed by the protocol:

  - Router discovery—How a host locates routers residing on an attached link.
  - Prefix discovery—How a host discovers address prefixes for destinations residing on an attached link. Nodes use prefixes to distinguish between destinations that reside on an attached link and those destinations that it can reach only through a router.
  - Parameter discovery—How a node learns various parameters (link parameters or Internet parameters) that it places in outgoing packets.
  - Address resolution—How a node uses only a destination IPv6 address to determine a link-layer address for destinations on an attached link.
  - Next-hop determination—The algorithm that a node uses for mapping an IPv6 destination address into a neighbor IPv6 address (either the next router hop or the destination itself) to which it plans to send traffic for the destination.
  - Neighbor unreachability detection—How a node determines that it can no longer reach a neighbor.
  - Duplicate address detection—How a node determines whether an address is already in use by another node.

- **Internet Control Message Protocol v6 (ICMPv6)**

  ICMP sends error messages and information messages related to IP operations. ICMPv6 defines additional error messages and informational messages specific to IPv6.

  There are four different ICMPv6 error messages:

- Destination Unreachable—A packet cannot be delivered due to an inherent problem with how it is being sent. Includes a code that indicates the nature of the problem that caused the packet not to be delivered
- Packet Too Big—Sent when a packet is too large to be delivered.
- Time Exceeded—A packet cannot be delivered because it has exceeded the hop count specified in the basic header hop-by-hop field.
- Parameter Problem—Indicates a problem with a field in the IPv6 header or extension headers that makes it impossible to process the packet.

ICMPv6 information messages are used for sharing the information required to implement various test, diagnostic, and support functions that are critical to the operation of IPv6. There are a total of eight different ICMPv6 informational messages:

- Echo Request—
- Echo Reply—
- Router Advertisement—
- Router Solicitation—
- Neighbor Advertisement—
- Neighbor Solicitation—
- Redirect—
- Router Renumbering—

- **Static routes for IPv6**

  Routing information can be configured statically. Whenever a route is configured statically, the routing information base (RIB) is updated with routes specified through the static route. These routes should be configured statically in the “routing-options” hierarchy. The following configuration is used for enabling static routes for IPv6:

```
interfaces {
   fe/0/1/0 {
      unit 0 {
         family inet6 {
            address fec0:0:0:3::1/64;
         }
      }
   }
}
   routing-options {
      rib inet6.0 {
         static {
            route fec0:0:0:4::/64 next-hop fec0:0:0:3::ffff;
         }
      }
   }
```

```
user@router> show route table inet6.0
```



```
inet6.0: 3 destination, 3 routes (3 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

fec0:0:0:3::/64  *[Direct/0] 00:01:34

     > via fe-0/1/0.0

fec:0:0:0:3::1/128  *[Local/0] 00:01:34

     Local

fec0:0:0:4::/64  *[Static/5] 00:01:34

     > to fec0:0:03:ffff via fe-0/1/0.0
```

Related Documentation

- IPv6 Overview on page 26
- *Understanding Dual Stacking of IPv4 and IPv6 Unicast Addresses*
- *IS-IS Overview*
- *OSPF Overview*
- *ICMP Router Discovery Overview*
- *MPLS Overview for ACX Series Universal Metro Routers*
- *Configuring Junos OS for IPv6 Path MTU Discovery*
- *IPv6 Neighbor Discovery Overview*

PART 2

Monitoring and Troubleshooting

- Monitoring Networks on page 43
- Troubleshooting Network Issues on page 49

CHAPTER 3

Monitoring Networks

- Example: Tracing Global Routing Protocol Operations on page 43

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 43
- Overview on page 43
- Configuration on page 44
- Verification on page 47

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.

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 *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
```

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
```

```
*** 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*
- CLI Explorer

CHAPTER 4

Troubleshooting Network Issues

- Working with Problems on Your Network on page 49
- Isolating a Broken Network Connection on page 50
- Identifying the Symptoms of a Broken Network Connection on page 51
- Isolating the Causes of a Network Problem on page 52
- Taking Appropriate Action for Resolving the Network Problem on page 53
- Evaluating the Solution to Check Whether the Network Problem Is Resolved on page 54

Working with Problems on Your Network

Problem **Description:** This checklist provides links to troubleshooting basics, an example network, and includes a summary of the commands you might use to diagnose problems with the router and network.

Solution

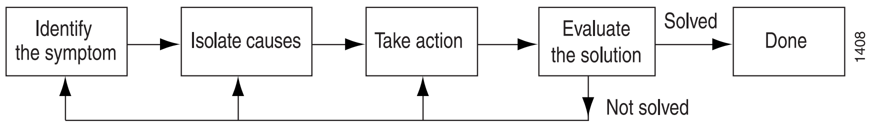
Table 6: Checklist for Working with Problems on Your Network

<table>
<tr><th>Tasks</th><th>Command or Action</th></tr>
<tr><td colspan="2">“Isolating a Broken Network Connection” on page 50</td></tr>
<tr><td>1. Identifying the Symptoms of a Broken Network Connection on page 51</td><td>ping (ip-address | hostname)
show route (ip-address | hostname)
traceroute (ip-address | hostname)</td></tr>
<tr><td>2. Isolating the Causes of a Network Problem on page 52</td><td>show < configuration | interfaces | protocols | route ></td></tr>
<tr><td>3. Taking Appropriate Action for Resolving the Network Problem on page 53</td><td>[edit]
delete routing options static route destination-prefix
commit and-quit
show route destination-prefix</td></tr>
<tr><td>4. Evaluating the Solution to Check Whether the Network Problem Is Resolved on page 54</td><td>show route (ip-address | hostname)
ping (ip-address | hostname) count 3
traceroute (ip-address | hostname)</td></tr>
</table>

Isolating a Broken Network Connection

By applying the standard four-step process illustrated in Figure 7 on page 50, you can isolate a failed node in the network. Note that the functionality described in this section is not supported in versions 15.1X49, 15.1X49-D30, or 15.1X49-D40.

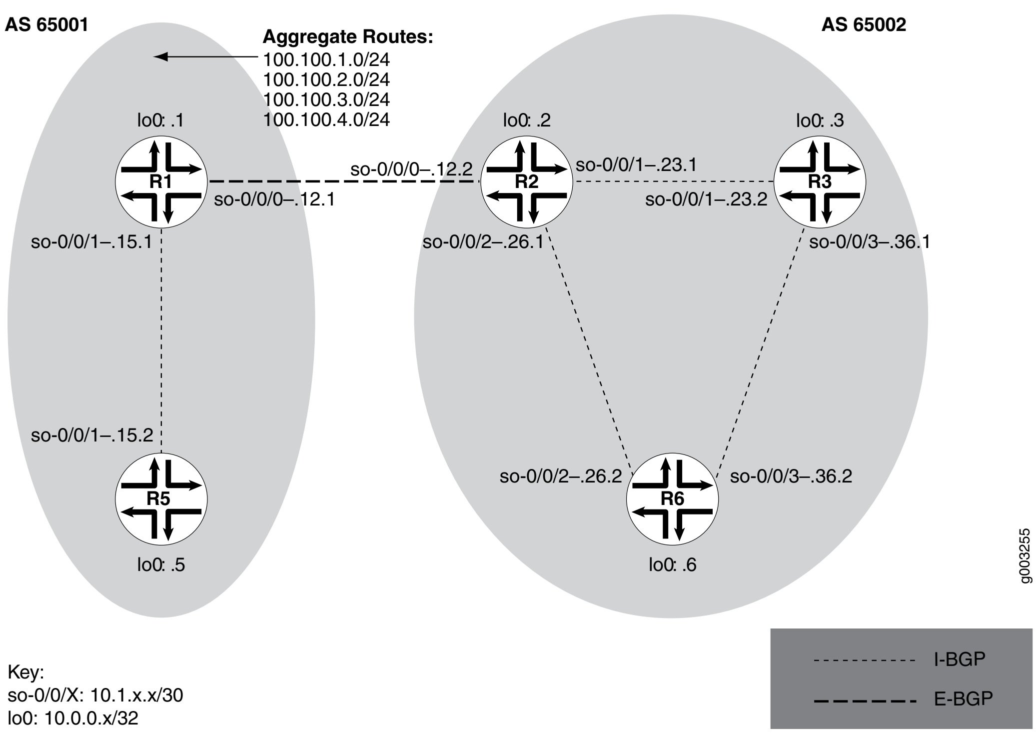
Figure 7: Process for Diagnosing Problems in Your Network



Before you embark on the four-step process, however, it is important that you are prepared for the inevitable problems that occur on all networks. While you might find a solution to a problem by simply trying a variety of actions, you can reach an appropriate solution more quickly if you are systematic in your approach to the maintenance and monitoring of your network. To prepare for problems on your network, understand how the network functions under normal conditions, have records of baseline network activity, and carefully observe the behavior of your network during a problem situation.

Figure 8 on page 50 shows the network topology used in this topic to illustrate the process of diagnosing problems in a network.

Figure 8: Network with a Problem



The network in Figure 8 on page 50 consists of two autonomous systems (ASs). AS 65001 includes two routers, and AS 65002 includes three routers. The border router (**R1**) in AS 65001 announces aggregated prefixes **100.100/24** to the AS 65002 network. The problem in this network is that **R6** does not have access to **R5** because of a loop between **R2** and **R6**.

To isolate a failed connection in your network, follow the steps in these topics:

- Identifying the Symptoms of a Broken Network Connection on page 51
- Isolating the Causes of a Network Problem on page 52
- Taking Appropriate Action for Resolving the Network Problem on page 53
- Taking Appropriate Action for Resolving the Network Problem on page 53
- Evaluating the Solution to Check Whether the Network Problem Is Resolved on page 54

Identifying the Symptoms of a Broken Network Connection

Problem **Description:** The symptoms of a problem in your network are usually quite obvious, such as the failure to reach a remote host.

Solution To identify the symptoms of a problem on your network, start at one end of your network and follow the routes to the other end, entering all or one of the following Junos OS command-line interfaces (CLI) operational mode commands:

```
user@host> ping (ip-address | host-name)
user@host> show route (ip-address | host-name)
user@host> traceroute (ip-address | host-name)
```

Sample Output

```
user@R6> ping 10.0.0.5
PING 10.0.0.5 (10.0.0.5): 56 data bytes
36 bytes from 10.1.26.1: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 e2db   0 0000  01  01 a8c6 10.1.26.2  10.0.0.5

36 bytes from 10.1.26.1: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 e2de   0 0000  01  01 a8c3 10.1.26.2  10.0.0.5

36 bytes from 10.1.26.1: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 e2e2   0 0000  01  01 a8bf 10.1.26.2  10.0.0.5

^C
--- 10.0.0.5 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss

user@R6> show route 10.0.0.5

inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.5/32        *[IS-IS/165] 00:02:39, metric 10
                     > to 10.1.26.1  via so-0/0/2.0
```



```
user@R6> traceroute 10.0.0.5
traceroute to 10.0.0.5 (10.0.0.5), 30 hops max, 40 byte packets
 1  10.1.26.1 (10.1.26.1)  0.649 ms  0.521 ms  0.490 ms
 2  10.1.26.2 (10.1.26.2)  0.521 ms  0.537 ms  0.507 ms
 3  10.1.26.1 (10.1.26.1)  0.523 ms  0.536 ms  0.514 ms
 4  10.1.26.2 (10.1.26.2)  0.528 ms  0.551 ms  0.523 ms
 5  10.1.26.1 (10.1.26.1)  0.531 ms  0.550 ms  0.524 ms
```

Meaning

The sample output shows an unsuccessful **ping** command in which the packets are being rejected because the time to live is exceeded. The output for the **show route** command shows the interface (**10.1.26.1**) that you can examine further for possible problems. The **traceroute** command shows the loop between **10.1.26.1** (**R2**) and **10.1.26.2** (**R6**), as indicated by the continuous repetition of the two interface addresses.

Isolating the Causes of a Network Problem

Problem

Description: A particular symptom can be the result of one or more causes. Narrow down the focus of your search to find each individual cause of the unwanted behavior.

Solution

To isolate the cause of a particular problem, enter one or all of the following Junos OS CLI operational mode command:

```
user@host> show < configuration | bgp | interfaces | isis | ospf | route >
```

Your particular problem may require the use of more than just the commands listed above. See the appropriate command reference for a more exhaustive list of commonly used operational mode commands.

Sample Output

```
user@R6> show interfaces terse
Interface               Admin Link Proto Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    up   inet  10.1.56.2/30
                                   iso
so-0/0/2                up    up
so-0/0/2.0              up    up   inet  10.1.26.2/30
                                   iso
so-0/0/3                up    up
so-0/0/3.0              up    up   inet  10.1.36.2/30
                                   iso
[...Output truncated...]
```

The following sample output is from **R2**:

```
user@R2> show route 10.0.0.5

inet.0: 22 destinations, 25 routes (22 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both

10.0.0.5/32          *[Static/5] 00:16:21
                      > to 10.1.26.2 via so-0/0/2.0
                      [BGP/170] 3d 20:23:35, MED 5, localpref 100
                        AS path: 65001 I
                      > to 10.1.12.1 via so-0/0/0.0
```

Meaning

The sample output shows that all interfaces on **R6** are up. The output from **R2** shows that a static route [**Static/5**] configured on **R2** points to **R6** (**10.1.26.2**) and is the preferred route to **R5** because of its low preference value. However, the route is looping from **R2** to **R6**, as indicated by the missing reference to **R5** (**10.1.15.2**).

Taking Appropriate Action for Resolving the Network Problem

Problem **Description:** The appropriate action depends on the type of problem you have isolated. In this example, a static route configured on **R2** is deleted from the [**routing-options**] hierarchy level. Other appropriate actions might include the following:

Solution

- Check the local router's configuration and edit it if appropriate.
- Troubleshoot the intermediate router.
- Check the remote host configuration and edit it if appropriate.
- Troubleshoot routing protocols.
- Identify additional possible causes.

To resolve the problem in this example, enter the following Junos OS CLI commands:

```
[edit]
user@R2# delete routing-options static route destination-prefix
user@R2# commit and-quit
user@R2# show route destination-prefix
```

Sample Output

```
[edit]
user@R2# delete routing-options static route 10.0.0.5/32

[edit]
user@R2# commit and-quit
commit complete
Exiting configuration mode

user@R2> show route 10.0.0.5

inet.0: 22 destinations, 24 routes (22 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.0.0.5/32          *[BGP/170] 3d 20:26:17, MED 5, localpref 100
                        AS path: 65001 I
                      > to 10.1.12.1 via so-0/0/0.0
```

Meaning

The sample output shows the static route deleted from the [**routing-options**] hierarchy and the new configuration committed. The output for the **show route** command now shows the BGP route as the preferred route, as indicated by the asterisk (*).

Evaluating the Solution to Check Whether the Network Problem Is Resolved

Problem

Description: If the problem is solved, you are finished. If the problem remains or a new problem is identified, start the process over again.

You can address possible causes in any order. In relation to the network in "Isolating a Broken Network Connection" on page 50, we chose to work from the local router toward the remote router, but you might start at a different point, particularly if you have reason to believe that the problem is related to a known issue, such as a recent change in configuration.

Solution

To evaluate the solution, enter the following Junos OS CLI commands:

```
user@host> show route (ip-address |host-name)
user@host> ping (ip-address | host-name)
user@host> traceroute (ip-address | host-name)
```

Sample Output

```
user@R6> show route 10.0.0.5

inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.5/32         *[BGP/170] 00:01:35, MED 5, localpref 100, from 10.0.0.2
                      AS path: 65001 I
                    > to 10.1.26.1 via so-0/0/2.0

user@R6> ping 10.0.0.5
PING 10.0.0.5 (10.0.0.5): 56 data bytes
64 bytes from 10.0.0.5: icmp_seq=0 ttl=253 time=0.866 ms
64 bytes from 10.0.0.5: icmp_seq=1 ttl=253 time=0.837 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=253 time=0.796 ms
^C
--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.796/0.833/0.866/0.029 ms

user@R6> traceroute 10.0.0.5
traceroute to 10.0.0.5 (10.0.0.5), 30 hops max, 40 byte packets
 1  10.1.26.1 (10.1.26.1)  0.629 ms  0.538 ms  0.497 ms
```



```
2  10.1.12.1 (10.1.12.1)  0.534 ms  0.538 ms  0.510 ms
3  10.0.0.5 (10.0.0.5)  0.776 ms  0.705 ms  0.672 ms
```

Meaning

The sample output shows that there is now a connection between **R6** and **R5.** The **show route** command shows that the BGP route to **R5** is preferred, as indicated by the asterisk (*). The **ping** command is successful and the **traceroute** command shows that the path from **R6** to **R5** is through **R2** (**10.1.26.1**), and then through **R1** (**10.1.12.1**).

PART 3

Configuration Statements and Operational Commands

- Operational Commands on page 59

CHAPTER 5

Operational Commands

- show | display rfc5952

show | display rfc5952

Syntax

show | display rfc5952

Release Information

Command introduced before Junos OS Release 11.4.

Description

Display IPv6 addresses as per RFC 5952 specifications. For RFC information, go to: http://tools.ietf.org/html/rfc5952.

Required Privilege Level

View

Related Documentation

- *show*

Sample Output

user@host> **show configuration interfaces ge-0/0/0 |display rfc5952**

```
unit 0 {
    family inet6 {
        address 2012::2:1/64;
        address 2001::1111:2222:0:0:1/96;
    }
}
```