BGP on QFX Series

Release
13.2X52

Published: 2014-07-15

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

BGP on QFX Series
13.2X52

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

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

Table of Contents

List of Figures

List of Tables

About the Documentation

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 http://www.juniper.net/techpubs/.

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 http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- QFabric System

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 the *CLI User Guide*.

Documentation Conventions

Table 1 on page xv 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 xv 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**

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

Convention	Description	Examples
`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.	
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**.

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

Convention	Description	Examples
> (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, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page at the Juniper Networks Technical Documentation site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at https://www.juniper.net/cgi-bin/docbugreport/.
- 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 JNASC 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 http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

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

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/

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

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

CHAPTER 1

BGP Overview

Understanding BGP

BGP is an exterior gateway protocol (EGP) that is used to exchange routing information among routers in different autonomous systems (ASs). BGP routing information includes the complete route to each destination. BGP uses the routing information to maintain a database of network reachability information, which it exchanges with other BGP systems. BGP uses the network reachability information to construct a graph of AS connectivity, which enables BGP to remove routing loops and enforce policy decisions at the AS level.

Multiprotocol BGP (MBGP) extensions enable BGP to support IP version 6 (IPv6). MBGP defines the attributes MP_REACH_NLRI and MP_UNREACH_NLRI, which are used to carry IPv6 reachability information. Network layer reachability information (NLRI) update messages carry IPv6 address prefixes of feasible routes.

BGP allows for policy-based routing. You can use routing policies to choose among multiple paths to a destination and to control the redistribution of routing information.

BGP uses TCP as its transport protocol, using port 179 for establishing connections. Running over a reliable transport protocol eliminates the need for BGP to implement update fragmentation, retransmission, acknowledgment, and sequencing.

The Junos OS routing protocol software supports BGP version 4. This version of BGP adds support for Classless Interdomain Routing (CIDR), which eliminates the concept of network classes. Instead of assuming which bits of an address represent the network by looking at the first octet, CIDR allows you to explicitly specify the number of bits in the network address, thus providing a means to decrease the size of the routing tables. BGP version 4 also supports aggregation of routes, including the aggregation of AS paths.

This section discusses the following topics:

Autonomous Systems

An *autonomous system* (AS) is a set of routers that are under a single technical administration and normally use a single interior gateway protocol and a common set of metrics to propagate routing information within the set of routers. To other ASs, an AS appears to have a single, coherent interior routing plan and presents a consistent picture of what destinations are reachable through it.

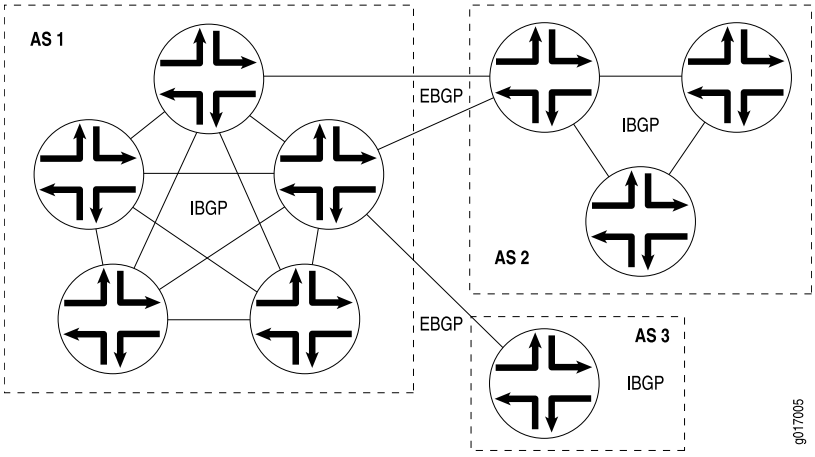
AS Paths and Attributes

The routing information that BGP systems exchange includes the complete route to each destination, as well as additional information about the route. The route to each destination is called the *AS path*, and the additional route information is included in *path attributes*. BGP uses the AS path and the path attributes to completely determine the network topology. Once BGP understands the topology, it can detect and eliminate routing loops and select among groups of routes to enforce administrative preferences and routing policy decisions.

External and Internal BGP

BGP supports two types of exchanges of routing information: exchanges among different ASs and exchanges within a single AS. When used among ASs, BGP is called *external BGP* (EBGP) and BGP sessions perform *inter-AS routing*. When used within an AS, BGP is called *internal BGP* (IBGP) and BGP sessions perform *intra-AS routing*. Figure 1 on page 4 illustrates ASs, IBGP, and EBGP.

Figure 1: ASs, EBGP, and IBGP



A BGP system shares network reachability information with adjacent BGP systems, which are referred to as *neighbors* or *peers*.

BGP systems are arranged into *groups*. In an IBGP group, all peers in the group—called *internal peers*—are in the same AS. Internal peers can be anywhere in the local AS and do not have to be directly connected to one another. Internal groups use routes from an

IGP to resolve forwarding addresses. They also propagate external routes among all other internal routers running IBGP, computing the next hop by taking the BGP next hop received with the route and resolving it using information from one of the interior gateway protocols.

In an EBGP group, the peers in the group—called *external peers*—are in different ASs and normally share a subnet. In an external group, the next hop is computed with respect to the interface that is shared between the external peer and the local router.

Multiple Instances of BGP

You can configure multiple instances of BGP at the following hierarchy levels:

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

Multiple instances of BGP are primarily used for Layer 3 VPN support.

IGP peers and external BGP (EBGP) peers (both nonmultihop and multihop) are all supported for routing instances. BGP peering is established over one of the interfaces configured under the **routing-instances** hierarchy.

NOTE: When a BGP neighbor sends BGP messages to the local routing device, the incoming interface on which these messages are received must be configured in the same routing instance that the BGP neighbor configuration exists in. This is true for neighbors that are a single hop away or multiple hops away.

Routes learned from the BGP peer are added to the **instance-name.inet.0** table by default. You can configure import and export policies to control the flow of information into and out of the instance routing table.

For Layer 3 VPN support, configure BGP on the provider edge (PE) router to receive routes from the customer edge (CE) router and to send the instances' routes to the CE router if necessary. You can use multiple instances of BGP to maintain separate per-site forwarding tables for keeping VPN traffic separate on the PE router.

You can configure import and export policies that allow the service provider to control and rate-limit traffic to and from the customer.

You can configure an EBGP multihop session for a VRF routing instance. Also, you can set up the EBGP peer between the PE and CE routers by using the loopback address of the CE router instead of the interface addresses.

Related Documentation

- BGP Routes Overview on page 6
- BGP Messages Overview on page 7

BGP Routes Overview

A BGP route is a destination, described as an IP address prefix, and information that describes the path to the destination.

The following information describes the path:

- AS path, which is a list of numbers of the ASs that a route passes through to reach the local router. The first number in the path is that of the last AS in the path—the AS closest to the local router. The last number in the path is the AS farthest from the local router, which is generally the origin of the path.
- Path attributes, which contain additional information about the AS path that is used in routing policy.

BGP peers advertise routes to each other in update messages.

BGP stores its routes in the Junos OS routing table (**inet.0**). The routing table stores the following information about BGP routes:

- Routing information learned from update messages received from peers
- Local routing information that BGP applies to routes because of local policies
- Information that BGP advertises to BGP peers in update messages

For each prefix in the routing table, the routing protocol process selects a single best path, called the active path. Unless you configure BGP to advertise multiple paths to the same destination, BGP advertises only the active path.

The BGP router that first advertises a route assigns it one of the following values to identify its origin. During route selection, the lowest origin value is preferred.

- **0**—The router originally learned the route through an IGP (OSPF, IS-IS, or a static route).
- **1**—The router originally learned the route through an EGP (most likely BGP).
- **2**—The route's origin is unknown.

Related Documentation

- Understanding BGP Path Selection on page 197
- Example: Advertising Multiple Paths in BGP on page 249

BGP Messages Overview

All BGP messages have the same fixed-size header, which contains a marker field that is used for both synchronization and authentication, a length field that indicates the length of the packet, and a type field that indicates the message type (for example, open, update, notification, keepalive, and so on).

This section discusses the following topics:

Open Messages

After a TCP connection is established between two BGP systems, they exchange BGP open messages to create a BGP connection between them. Once the connection is established, the two systems can exchange BGP messages and data traffic.

Open messages consist of the BGP header plus the following fields:

- Version—The current BGP version number is 4.
- Local AS number—You configure this by including the **autonomous-system** statement at the **[edit routing-options]** or **[edit logical-systems *logical-system-name* routing-options]** hierarchy level.
- Hold time—Proposed hold-time value. You configure the local hold time with the BGP **hold-time** statement.
- BGP identifier—IP address of the BGP system. This address is determined when the system starts and is the same for every local interface and every BGP peer. You can configure the BGP identifier by including the **router-id** statement at the **[edit routing-options]** or **[edit logical-systems *logical-system-name* routing-options]** hierarchy level. By default, BGP uses the IP address of the first interface it finds in the router.
- Parameter field length and the parameter itself—These are optional fields.

Update Messages

BGP systems send update messages to exchange network reachability information. BGP systems use this information to construct a graph that describes the relationships among all known ASs.

Update messages consist of the BGP header plus the following optional fields:

- Unfeasible routes length—Length of the withdrawn routes field
- Withdrawn routes—IP address prefixes for the routes being withdrawn from service because they are no longer deemed reachable

- Total path attribute length—Length of the path attributes field; it lists the path attributes for a feasible route to a destination
- Path attributes—Properties of the routes, including the path origin, the multiple exit discriminator (MED), the originating system's preference for the route, and information about aggregation, communities, confederations, and route reflection
- Network layer reachability information (NLRI)—IP address prefixes of feasible routes being advertised in the update message

Keepalive Messages

BGP systems exchange keepalive messages to determine whether a link or host has failed or is no longer available. Keepalive messages are exchanged often enough so that the hold timer does not expire. These messages consist only of the BGP header.

Notification Messages

BGP systems send notification messages when an error condition is detected. After the message is sent, the BGP session and the TCP connection between the BGP systems are closed. Notification messages consist of the BGP header plus the error code and subcode, and data that describes the error.

Related Documentation

- Understanding BGP on page 3
- BGP Routes Overview on page 6

PART 2

Configuration

CHAPTER 2

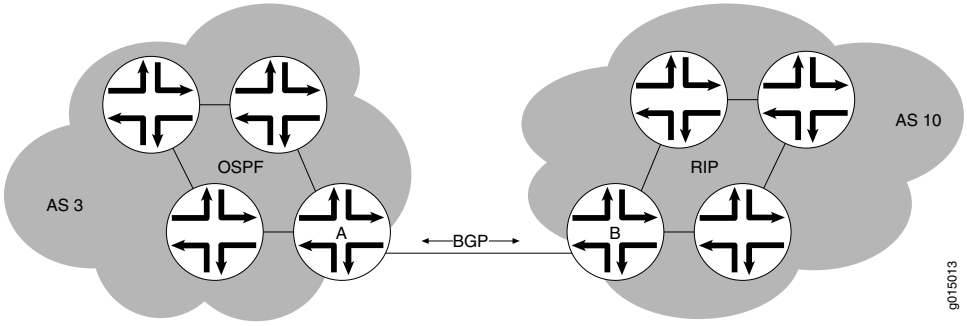
Basic BGP Configuration

Examples: Configuring External BGP Peering

Understanding External BGP Peering Sessions

To establish point-to-point connections between peer autonomous systems (ASs), you configure a BGP session on each interface of a point-to-point link. Generally, such sessions are made at network exit points with neighboring hosts outside the AS. Figure 2 on page 11 shows an example of a BGP peering session.

Figure 2: BGP Peering Session



In Figure 2 on page 11, Router A is a gateway router for AS 3, and Router B is a gateway router for AS 10. For traffic internal to either AS, an interior gateway protocol (IGP) is used (OSPF, for instance). To route traffic between peer ASs, a BGP session is used.

You arrange BGP routing devices into groups of peers. Different peer groups can have different group types, AS numbers, and route reflector cluster identifiers.

To define a BGP group that recognizes only the specified BGP systems as peers, statically configure all the system's peers by including one or more **neighbor** statements. The peer neighbor's address can be either an IPv6 or IPv4 address.

As the number of external BGP (EBGP) groups increases, the ability to support a large number of BGP sessions might become a scaling issue. The preferred way to configure a large number of BGP neighbors is to configure a few groups consisting of multiple neighbors per group. Supporting fewer EBGP groups generally scales better than supporting a large number of EBGP groups. This becomes more evident in the case of hundreds of EBGP groups when compared with a few EBGP groups with multiple peers in each group.

After the BGP peers are established, BGP routes are not automatically advertised by the BGP peers. At each BGP-enabled device, policy configuration is required to export the local, static, or IGP-learned routes into the BGP RIB and then advertise them as BGP routes to the other peers. BGP's advertisement policy, by default, does not advertise any non-BGP routes (such as local routes) to peers.

Example: Configuring External BGP Point-to-Point Peer Sessions

This example shows how to configure BGP point-to-point peer sessions.

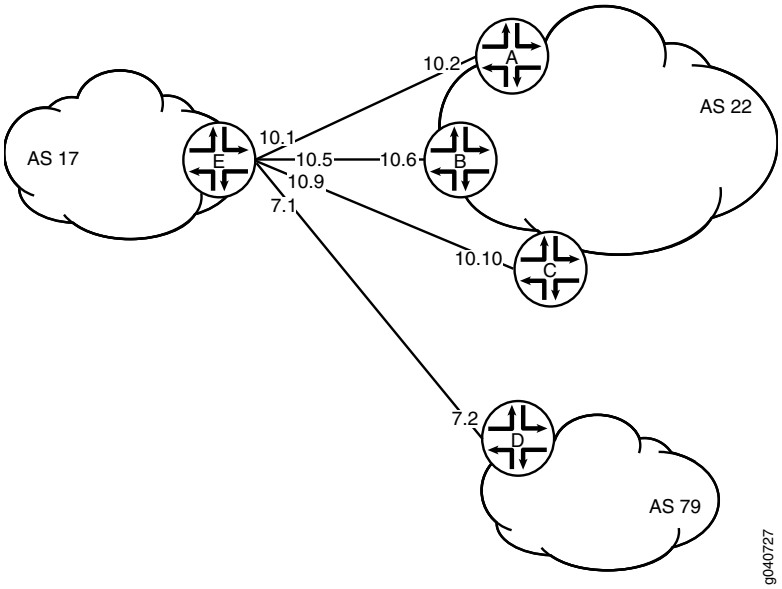
Requirements

Before you begin, if the default BGP policy is not adequate for your network, configure routing policies to filter incoming BGP routes and to advertise BGP routes.

Overview

Figure 3 on page 13 shows a network with BGP peer sessions. In the sample network, Device E in AS 17 has BGP peer sessions to a group of peers called **external-peers**. Peers A, B, and C reside in AS 22 and have IP addresses 10.10.10.2, 10.10.10.6, and 10.10.10.10. Peer D resides in AS 79, at IP address 10.21.7.2. This example shows the configuration on Device E.

Figure 3: Typical Network with BGP Peer Sessions



Configuration

CLI Quick Configuration

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

```
set interfaces ge-1/2/0 unit 0 description to-A
set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/30
set interfaces ge-0/0/1 unit 5 description to-B
set interfaces ge-0/0/1 unit 5 family inet address 10.10.10.5/30
set interfaces ge-0/1/0 unit 9 description to-C
set interfaces ge-0/1/0 unit 9 family inet address 10.10.10.9/30
set interfaces ge-1/2/1 unit 21 description to-D
set interfaces ge-1/2/1 unit 21 family inet address 10.21.7.1/30
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 22
set protocols bgp group external-peers neighbor 10.10.10.2
set protocols bgp group external-peers neighbor 10.10.10.6
set protocols bgp group external-peers neighbor 10.10.10.10
set protocols bgp group external-peers neighbor 10.21.7.2 peer-as 79
set routing-options autonomous-system 17
```

Step-by-Step Procedure

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

To configure the BGP peer sessions:

1. Configure the interfaces to Peers A, B, C, and D.

   [edit interfaces]

```
user@E# set ge-1/2/0 unit 0 description to-A
user@E# set ge-1/2/0 unit 0 family inet address 10.10.10.1/30
user@E# set ge-0/0/1 unit 5 description to-B
user@E# set ge-0/0/1 unit 5 family inet address 10.10.10.5/30
user@E# set ge-0/1/0 unit 9 description to-C
user@E# set ge-0/1/0 unit 9 family inet address 10.10.10.9/30
user@E# set ge-1/2/1 unit 21 description to-D
user@E# set ge-1/2/1 unit 21 family inet address 10.21.7.1/30
```

2. Set the autonomous system (AS) number.

```
[edit routing-options]
user@E# set autonomous-system 17
```

3. Create the BGP group, and add the external neighbor addresses.

```
[edit protocols bgp group external-peers]
user@E# set neighbor 10.10.10.2
user@E# set neighbor 10.10.10.6
user@E# set neighbor 10.10.10.10
```

4. Specify the autonomous system (AS) number of the external AS.

```
[edit protocols bgp group external-peers]
user@E# set peer-as 22
```

5. Add Peer D, and set the AS number at the individual neighbor level.

   The neighbor configuration overrides the group configuration. So, while **peer-as 22** is set for all the other neighbors in the group, **peer-as 79** is set for neighbor 10.21.7.2.

```
[edit protocols bgp group external-peers]
user@E# set neighbor 10.21.7.2 peer-as 79
```

6. Set the peer type to external BGP (EBGP).

```
[edit protocols bgp group external-peers]
user@E# set type external
```

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

```
[edit]
user@E# show interfaces
ge-1/2/0 {
    unit 0 {
        description to-A;
        family inet {
            address 10.10.10.1/30;
        }
    }
}
ge-0/0/1 {
    unit 5 {
        description to-B;
        family inet {
            address 10.10.10.5/30;
        }
```

```
    }
  }
  ge-0/1/0 {
    unit 9 {
      description to-C;
      family inet {
        address 10.10.10.9/30;
      }
    }
  }
  ge-1/2/1 {
    unit 21 {
      description to-D;
      family inet {
        address 10.21.7.1/30;
      }
    }
  }

  [edit]
  user@E# show protocols
  bgp {
    group external-peers {
      type external;
      peer-as 22;
      neighbor 10.10.10.2;
      neighbor 10.10.10.6;
      neighbor 10.10.10.10;
      neighbor 10.21.7.2 {
        peer-as 79;
      }
    }
  }

  [edit]
  user@E# show routing-options
  autonomous-system 17;
```

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

Verification

Confirm that the configuration is working properly.

- Verifying BGP Neighbors on page 15
- Verifying BGP Groups on page 18
- Verifying BGP Summary Information on page 18

Verifying BGP Neighbors

Purpose Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

Action From operational mode, run the **show bgp neighbor** command.

```
user@E> show bgp neighbor
```

```
Peer: 10.10.10.2+179 AS 22     Local: 10.10.10.1+65406 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.10.10.2       Local ID: 10.10.10.1       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  Local Interface: ge-1/2/0.0
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 10   Sent 6    Checked 1
  Input messages:  Total 8522   Updates 1       Refreshes 0     Octets 161922
  Output messages: Total 8433   Updates 0       Refreshes 0     Octets 160290
  Output Queue[0]: 0

Peer: 10.10.10.6+54781 AS 22   Local: 10.10.10.5+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.10.10.6       Local ID: 10.10.10.1       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  Local Interface: ge-0/0/1.5
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
```

```
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 12   Sent 6    Checked 33
  Input messages:  Total 8527   Updates 1       Refreshes 0     Octets 162057
  Output messages: Total 8430   Updates 0       Refreshes 0     Octets 160233
  Output Queue[0]: 0

Peer: 10.10.10.10+55012 AS 22  Local: 10.10.10.9+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.10.10.10      Local ID: 10.10.10.1       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 2
  BFD: disabled, down
  Local Interface: fe-0/1/0.9
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 15   Sent 6    Checked 37
  Input messages:  Total 8527   Updates 1       Refreshes 0     Octets 162057
  Output messages: Total 8429   Updates 0       Refreshes 0     Octets 160214
  Output Queue[0]: 0

Peer: 10.21.7.2+61867 AS 79    Local: 10.21.7.1+179 AS 17
  Type: External    State: Established    Flags: <ImportEval Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.21.7.2        Local ID: 10.10.10.1       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 3
  BFD: disabled, down
  Local Interface: ge-1/2/1.21
```

```
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 79)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 28   Sent 24   Checked 47
  Input messages:  Total 8521   Updates 1       Refreshes 0     Octets 161943
  Output messages: Total 8427   Updates 0       Refreshes 0     Octets 160176
  Output Queue[0]: 0
```

Verifying BGP Groups

Purpose Verify that the BGP groups are configured correctly.

Action From operational mode, run the **show bgp group** command.

```
user@E> show bgp group
Group Type: External                              Local AS: 17
  Name: external-peers  Index: 0                  Flags: <>
  Holdtime: 0
  Total peers: 4        Established: 4
  10.10.10.2+179
  10.10.10.6+54781
  10.10.10.10+55012
  10.21.7.2+61867
  inet.0: 0/0/0/0

Groups: 1  Peers: 4    External: 4    Internal: 0    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
```

Verifying BGP Summary Information

Purpose Verify that the BGP configuration is correct.

Action From operational mode, run the **show bgp summary** command.

```
user@E> show bgp summary
Groups: 1 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.10.10.2               22       8559       8470       0       0 2d 16:12:56
```

```
0/0/0/0               0/0/0/0
10.10.10.6              22       8566       8468       0       0 2d 16:12:12
0/0/0/0               0/0/0/0
10.10.10.10             22       8565       8466       0       0 2d 16:11:31
0/0/0/0               0/0/0/0
10.21.7.2               79       8560       8465       0       0 2d 16:10:58
0/0/0/0               0/0/0/0
```

Example: Configuring External BGP on Logical Systems with IPv6 Interfaces

This example shows how to configure external BGP (EBGP) point-to-point peer sessions on logical systems with IPv6 interfaces.

Requirements

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

Overview

Junos OS supports EBGP peer sessions by means of IPv6 addresses. An IPv6 peer session can be configured when an IPv6 address is specified in the **neighbor** statement. This example uses EUI-64 to generate IPv6 addresses that are automatically applied to the interfaces. An EUI-64 address is an IPv6 address that uses the IEEE EUI-64 format for the interface identifier portion of the address (the last 64 bits).

NOTE: Alternatively, you can configure EBGP sessions using manually assigned 128-bit IPv6 addresses.

If you use 128-bit link-local addresses for the interfaces, you must include the **local-interface** statement. This statement is valid only for 128-bit IPv6 link-local addresses and is mandatory for configuring an IPv6 EBGP link-local peer session.

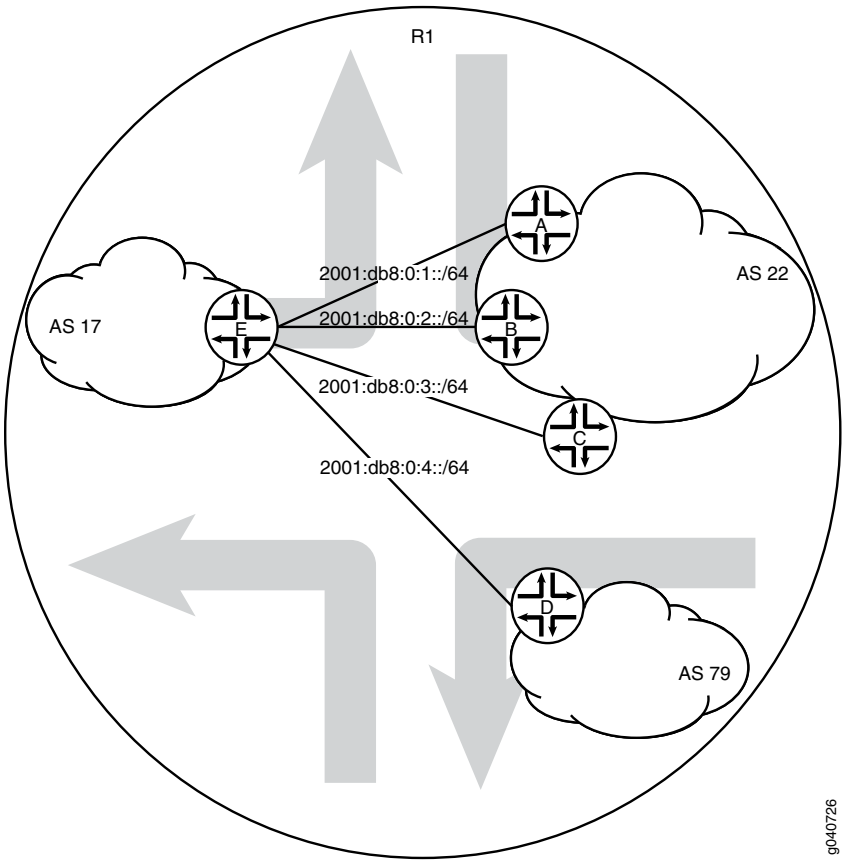
Configuring EBGP peering using link-local addresses is only applicable for directly connected interfaces. There is no support for multihop peering.

After your interfaces are up, you can use the **show interfaces terse** command to view the EUI-64-generated IPv6 addresses on the interfaces. You must use these generated addresses in the BGP **neighbor** statements. This example demonstrates the full end-to-end procedure.

In this example, Frame Relay interface encapsulation is applied to the logical tunnel (**lt**) interfaces. This is a requirement because only Frame Relay encapsulation is supported when IPv6 addresses are configured on the **lt** interfaces.

Figure 4 on page 20 shows a network with BGP peer sessions. In the sample network, Router R1 has five logical systems configured. Device E in autonomous system (AS) 17 has BGP peer sessions to a group of peers called **external-peers**. Peers A, B, and C reside in AS 22. This example shows the step-by-step configuration on Logical System A and Logical System E.

Figure 4: Typical Network with BGP Peer Sessions



Configuration

CLI Quick Configuration

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

Device A

```
set logical-systems A interfaces lt-0/1/0 unit 1 description to-E
set logical-systems A interfaces lt-0/1/0 unit 1 encapsulation frame-relay
set logical-systems A interfaces lt-0/1/0 unit 1 dlci 1
set logical-systems A interfaces lt-0/1/0 unit 1 peer-unit 25
set logical-systems A interfaces lt-0/1/0 unit 1 family inet6 address 2001:db8:0:1::/64
    eui-64
set logical-systems A interfaces lo0 unit 1 family inet6 address 2001:db8::1/128
set logical-systems A protocols bgp group external-peers type external
set logical-systems A protocols bgp group external-peers peer-as 17
```

```
set logical-systems A protocols bgp group external-peers neighbor
  2001:db8:0:1:2a0:a502:0:19da
set logical-systems A routing-options router-id 1.1.1.1
set logical-systems A routing-options autonomous-system 22
```

Device B

```
set logical-systems B interfaces lt-0/1/0 unit 6 description to-E
set logical-systems B interfaces lt-0/1/0 unit 6 encapsulation frame-relay
set logical-systems B interfaces lt-0/1/0 unit 6 dlci 6
set logical-systems B interfaces lt-0/1/0 unit 6 peer-unit 5
set logical-systems B interfaces lt-0/1/0 unit 6 family inet6 address 2001:db8:0:2::/64
  eui-64
set logical-systems B interfaces lo0 unit 2 family inet6 address 2001:db8::2/128
set logical-systems B protocols bgp group external-peers type external
set logical-systems B protocols bgp group external-peers peer-as 17
set logical-systems B protocols bgp group external-peers neighbor
  2001:db8:0:2:2a0:a502:0:5da
set logical-systems B routing-options router-id 2.2.2.2
set logical-systems B routing-options autonomous-system 22
```

Device C

```
set logical-systems C interfaces lt-0/1/0 unit 10 description to-E
set logical-systems C interfaces lt-0/1/0 unit 10 encapsulation frame-relay
set logical-systems C interfaces lt-0/1/0 unit 10 dlci 10
set logical-systems C interfaces lt-0/1/0 unit 10 peer-unit 9
set logical-systems C interfaces lt-0/1/0 unit 10 family inet6 address 2001:db8:0:3::/64
  eui-64
set logical-systems C interfaces lo0 unit 3 family inet6 address 2001:db8::3/128
set logical-systems C protocols bgp group external-peers type external
set logical-systems C protocols bgp group external-peers peer-as 17
set logical-systems C protocols bgp group external-peers neighbor
  2001:db8:0:3:2a0:a502:0:9da
set logical-systems C routing-options router-id 3.3.3.3
set logical-systems C routing-options autonomous-system 22
```

Device D

```
set logical-systems D interfaces lt-0/1/0 unit 7 description to-E
set logical-systems D interfaces lt-0/1/0 unit 7 encapsulation frame-relay
set logical-systems D interfaces lt-0/1/0 unit 7 dlci 7
set logical-systems D interfaces lt-0/1/0 unit 7 peer-unit 21
set logical-systems D interfaces lt-0/1/0 unit 7 family inet6 address 2001:db8:0:4::/64
  eui-64
set logical-systems D interfaces lo0 unit 4 family inet6 address 2001:db8::4/128
set logical-systems D protocols bgp group external-peers type external
set logical-systems D protocols bgp group external-peers peer-as 17
set logical-systems D protocols bgp group external-peers neighbor
  2001:db8:0:4:2a0:a502:0:15da
set logical-systems D routing-options router-id 4.4.4.4
set logical-systems D routing-options autonomous-system 79
```

Device E

```
set logical-systems E interfaces lt-0/1/0 unit 5 description to-B
set logical-systems E interfaces lt-0/1/0 unit 5 encapsulation frame-relay
set logical-systems E interfaces lt-0/1/0 unit 5 dlci 6
set logical-systems E interfaces lt-0/1/0 unit 5 peer-unit 6
set logical-systems E interfaces lt-0/1/0 unit 5 family inet6 address 2001:db8:0:2::/64
  eui-64
set logical-systems E interfaces lt-0/1/0 unit 9 description to-C
set logical-systems E interfaces lt-0/1/0 unit 9 encapsulation frame-relay
```

```
set logical-systems E interfaces lt-0/1/0 unit 9 dlci 10
set logical-systems E interfaces lt-0/1/0 unit 9 peer-unit 10
set logical-systems E interfaces lt-0/1/0 unit 9 family inet6 address 2001:db8:0:3::/64
  eui-64
set logical-systems E interfaces lt-0/1/0 unit 21 description to-D
set logical-systems E interfaces lt-0/1/0 unit 21 encapsulation frame-relay
set logical-systems E interfaces lt-0/1/0 unit 21 dlci 7
set logical-systems E interfaces lt-0/1/0 unit 21 peer-unit 7
set logical-systems E interfaces lt-0/1/0 unit 21 family inet6 address 2001:db8:0:4::/64
  eui-64
set logical-systems E interfaces lt-0/1/0 unit 25 description to-A
set logical-systems E interfaces lt-0/1/0 unit 25 encapsulation frame-relay
set logical-systems E interfaces lt-0/1/0 unit 25 dlci 1
set logical-systems E interfaces lt-0/1/0 unit 25 peer-unit 1
set logical-systems E interfaces lt-0/1/0 unit 25 family inet6 address 2001:db8:0:1::/64
  eui-64
set logical-systems E interfaces lo0 unit 5 family inet6 address 2001:db8::5/128
set logical-systems E protocols bgp group external-peers type external
set logical-systems E protocols bgp group external-peers peer-as 22
set logical-systems E protocols bgp group external-peers neighbor
  2001:db8:0:1:2a0:a502:0:1da
set logical-systems E protocols bgp group external-peers neighbor
  2001:db8:0:2:2a0:a502:0:6da
set logical-systems E protocols bgp group external-peers neighbor
  2001:db8:0:3:2a0:a502:0:ada
set logical-systems E protocols bgp group external-peers neighbor
  2001:db8:0:4:2a0:a502:0:7da peer-as 79
set logical-systems E routing-options router-id 5.5.5.5
set logical-systems E routing-options autonomous-system 17
```

Step-by-Step Procedure

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

To configure the BGP peer sessions:

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

   ```
   user@R1> show interfaces terse
   Interface               Admin Link Proto    Local                 Remote
   ...
   lt-0/1/0                up    up
   ...
   ```

2. On Logical System A, configure the interface encapsulation, peer-unit number, and DLCI to reach Logical System E.

   ```
   user@R1> set cli logical-system A
   Logical system: A
   [edit]
   user@R1:A> edit
   Entering configuration mode
   [edit]
   user@R1:A# edit interfaces
   [edit interfaces]
   user@R1:A# set lt-0/1/0 unit 1 encapsulation frame-relay
   ```

```
user@R1:A# set lt-0/1/0 unit 1 dlci 1
user@R1:A# set lt-0/1/0 unit 1 peer-unit 25
```

3. On Logical System A, configure the network address for the link to Peer E, and configure a loopback interface.

```
[edit interfaces]
user@R1:A# set lt-0/1/0 unit 1 description to-E
user@R1:A# set lt-0/1/0 unit 1 family inet6 address 2001:db8:0:1::/64 eui-64
user@R1:A# set lo0 unit 1 family inet6 address 2001:db8::1/128
```

4. On Logical System E, configure the interface encapsulation, peer-unit number, and DLCI to reach Logical System A.

```
user@R1> set cli logical-system E
Logical system: E
[edit]
user@R1:E> edit
Entering configuration mode
[edit]
user@R1:E# edit interfaces
[edit interfaces]
user@R1:E# set lt-0/1/0 unit 25 encapsulation frame-relay
user@R1:E# set lt-0/1/0 unit 25 dlci 1
user@R1:E# set lt-0/1/0 unit 25 peer-unit 1
```

5. On Logical System E, configure the network address for the link to Peer A, and configure a loopback interface.

```
[edit interfaces]
user@R1:E# set lt-0/1/0 unit 25 description to-A
user@R1:E# set lt-0/1/0 unit 25 family inet6 address 2001:db8:0:1::/64 eui-64
user@R1:E# set lo0 unit 5 family inet6 address 2001:db8::5/128
```

6. Run the **show interfaces terse** command to see the IPv6 addresses that are generated by EUI-64.

   The 2001 addresses are used in this example in the BGP **neighbor** statements.

NOTE: The fe80 addresses are link-local addresses and are not used in this example.

```
user@R1:A> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
Logical system: A

betsy@tp8:A> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
lt-0/1/0
lt-0/1/0.1              up    up   inet6    2001:db8:0:1:2a0:a502:0:1da/64
                                            fe80::2a0:a502:0:1da/64
lo0
lo0.1                   up    up   inet6    2001:db8::1
                                            fe80::2a0:a50f:fc56:1da

user@R1:E> show interfaces terse
```

```
Interface               Admin Link Proto    Local                 Remote
lt-0/1/0
lt-0/1/0.25             up    up   inet6    2001:db8:0:1:2a0:a502:0:19da/64

                                            fe80::2a0:a502:0:19da/64
lo0
lo0.5                   up    up   inet6    2001:db8::5
                                            fe80::2a0:a50f:fc56:1da
```

7. Repeat the interface configuration on the other logical systems.

Configuring the External BGP Sessions

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 BGP peer sessions:

1. On Logical System A, create the BGP group, and add the external neighbor address.

   ```
   [edit protocols bgp group external-peers]
   user@R1:A# set neighbor 2001:db8:0:1:2a0:a502:0:19da
   ```

2. On Logical System E, create the BGP group, and add the external neighbor address.

   ```
   [edit protocols bgp group external-peers]
   user@R1:E# set neighbor 2001:db8:0:1:2a0:a502:0:1da
   ```

3. On Logical System A, specify the autonomous system (AS) number of the external AS.

   ```
   [edit protocols bgp group external-peers]
   user@R1:A# set peer-as 17
   ```

4. On Logical System E, specify the autonomous system (AS) number of the external AS.

   ```
   [edit protocols bgp group external-peers]
   user@R1:E# set peer-as 22
   ```

5. On Logical System A, set the peer type to EBGP.

   ```
   [edit protocols bgp group external-peers]
   user@R1:A# set type external
   ```

6. On Logical System E, set the peer type to EBGP.

   ```
   [edit protocols bgp group external-peers]
   user@R1:E# set type external
   ```

7. On Logical System A, set the autonomous system (AS) number and router ID.

   ```
   [edit routing-options]
   user@R1:A# set router-id 1.1.1.1
   user@R1:A# set autonomous-system 22
   ```

8. On Logical System E, set the AS number and router ID.

   ```
   [edit routing-options]
   user@R1:E# set router-id 5.5.5.5
   ```

```
user@R1:E# set autonomous-system 17
```

9. Repeat these steps for Peers A, B, C, and D.

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

```
[edit]
user@R1# show logical-systems
A {
    interfaces {
        lt-0/1/0 {
            unit 1 {
                description to-E;
                encapsulation frame-relay;
                dlci 1;
                peer-unit 25;
                family inet6 {
                    address 2001:db8:0:1::/64 {
                        eui-64;
                    }
                }
            }
        }
        lo0 {
            unit 1 {
                family inet6 {
                    address 2001:db8::1/128;
                }
            }
        }
    }
    protocols {
        bgp {
            group external-peers {
                type external;
                peer-as 17;
                neighbor 2001:db8:0:1:2a0:a502:0:19da;
            }
        }
        routing-options {
            router-id 1.1.1.1;
            autonomous-system 22;
        }
    }
    B {
        interfaces {
            lt-0/1/0 {
                unit 6 {
                    description to-E;
                    encapsulation frame-relay;
                    dlci 6;
                    peer-unit 5;
                    family inet6 {
```

```
            address 2001:db8:0:2::/64 {
              eui-64;
            }
          }
        }
      }
      lo0 {
        unit 2 {
          family inet6 {
            address 2001:db8::2/128;
          }
        }
      }
    }
    protocols {
      bgp {
        group external-peers {
          type external;
          peer-as 17;
          neighbor 2001:db8:0:2:2a0:a502:0:5da;
        }
      }
      routing-options {
        router-id 2.2.2.2;
        autonomous-system 22;
      }
    }
    C {
      interfaces {
        lt-0/1/0 {
          unit 10 {
            description to-E;
            encapsulation frame-relay;
            dlci 10;
            peer-unit 9;
            family inet6 {
              address 2001:db8:0:3::/64 {
                eui-64;
              }
            }
          }
        }
        lo0 {
          unit 3 {
            family inet6 {
              address 2001:db8::3/128;
            }
          }
        }
      }
      protocols {
        bgp {
          group external-peers {
            type external;
            peer-as 17;
            neighbor 2001:db8:0:3:2a0:a502:0:9da;
```

```
        }
      }
    }
    routing-options {
      router-id 3.3.3.3;
      autonomous-system 22;
    }
  }
  D {
    interfaces {
      lt-0/1/0 {
        unit 7 {
          description to-E;
          encapsulation frame-relay;
          dlci 7;
          peer-unit 21;
          family inet6 {
            address 2001:db8:0:4::/64 {
              eui-64;
            }
          }
        }
      }
      lo0 {
        unit 4 {
          family inet6 {
            address 2001:db8::4/128;
          }
        }
      }
    }
    protocols {
      bgp {
        group external-peers {
          type external;
          peer-as 17;
          neighbor 2001:db8:0:4:2a0:a502:0:15da;
        }
      }
      routing-options {
        router-id 4.4.4.4;
        autonomous-system 79;
      }
    }
    E {
      interfaces {
        lt-0/1/0 {
          unit 5 {
            description to-B;
            encapsulation frame-relay;
            dlci 6;
            peer-unit 6;
            family inet6 {
              address 2001:db8:0:2::/64 {
                eui-64;
              }
```

```
        }
      }
      unit 9 {
        description to-C;
        encapsulation frame-relay;
        dlci 10;
        peer-unit 10;
        family inet6 {
          address 2001:db8:0:3::/64 {
            eui-64;
          }
        }
      }
      unit 21 {
        description to-D;
        encapsulation frame-relay;
        dlci 7;
        peer-unit 7;
        family inet6 {
          address 2001:db8:0:4::/64 {
            eui-64;
          }
        }
      }
      unit 25 {
        description to-A;
        encapsulation frame-relay;
        dlci 1;
        peer-unit 1;
        family inet6 {
          address 2001:db8:0:1::/64 {
            eui-64;
          }
        }
      }
    }
    lo0 {
      unit 5 {
        family inet6 {
          address 2001:db8::5/128;
        }
      }
    }
  }
  protocols {
    bgp {
      group external-peers {
        type external;
        peer-as 22;
        neighbor 2001:db8:0:1:2a0:a502:0:1da;
        neighbor 2001:db8:0:2:2a0:a502:0:6da;
        neighbor 2001:db8:0:3:2a0:a502:0:ada;
        neighbor 2001:db8:0:4:2a0:a502:0:7da {
          peer-as 79;
        }
      }
```

```
    }
  }
  routing-options {
    router-id 5.5.5.5;
    autonomous-system 17;
  }
}
```

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

Verification

Confirm that the configuration is working properly.

- Verifying BGP Neighbors on page 29
- Verifying BGP Groups on page 32
- Verifying BGP Summary Information on page 32
- Checking the Routing Table on page 32

Verifying BGP Neighbors

Purpose Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

Action From operational mode, run the **show bgp neighbor** command.

```
user@R1:E> show bgp neighbor
Peer: 2001:db8:0:1:2a0:a502:0:1da+54987 AS 22 Local:
2001:db8:0:1:2a0:a502:0:19da+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Open Message Error
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Error: 'Open Message Error' Sent: 20 Recv: 0
  Peer ID: 1.1.1.1         Local ID: 5.5.5.5          Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  Local Interface: lt-0/1/0.25
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
```

```
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 7    Sent 18   Checked 81
  Input messages:  Total 1611   Updates 1       Refreshes 0     Octets 30660
  Output messages: Total 1594   Updates 0       Refreshes 0     Octets 30356
  Output Queue[0]: 0

Peer: 2001:db8:0:2:2a0:a502:0:6da+179 AS 22 Local:
2001:db8:0:2:2a0:a502:0:5da+55502 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Open Message Error
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Error: 'Open Message Error' Sent: 26 Recv: 0
  Peer ID: 2.2.2.2         Local ID: 5.5.5.5           Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 2
  BFD: disabled, down
  Local Interface: lt-0/1/0.5
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 15   Sent 8    Checked 8
  Input messages:  Total 1610   Updates 1       Refreshes 0     Octets 30601
  Output messages: Total 1645   Updates 0       Refreshes 0     Octets 32417
  Output Queue[0]: 0

Peer: 2001:db8:0:3:2a0:a502:0:ada+55983 AS 22 Local:
2001:db8:0:3:2a0:a502:0:9da+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 3.3.3.3         Local ID: 5.5.5.5           Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 3
  BFD: disabled, down
  Local Interface: lt-0/1/0.9
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
```

```
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 22)
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 21   Sent 21   Checked 67
  Input messages:  Total 1610   Updates 1       Refreshes 0     Octets 30641
  Output messages: Total 1587   Updates 0       Refreshes 0     Octets 30223
  Output Queue[0]: 0

Peer: 2001:db8:0:4:2a0:a502:0:7da+49255 AS 79 Local:
2001:db8:0:4:2a0:a502:0:15da+179 AS 17
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 4.4.4.4         Local ID: 5.5.5.5           Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  Local Interface: lt-0/1/0.21
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer supports 4 byte AS extension (peer-as 79)
  Peer does not support Addpath
  Table inet6.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 6    Sent 17   Checked 25
  Input messages:  Total 1615   Updates 1       Refreshes 0     Octets 30736
  Output messages: Total 1593   Updates 0       Refreshes 0     Octets 30337
  Output Queue[0]: 0
```

Meaning IPv6 unicast network layer reachability information (NLRI) is being exchanged between the neighbors.

Verifying BGP Groups

Purpose Verify that the BGP groups are configured correctly.

Action From operational mode, run the **show bgp group** command.

```
user@R1:E> show bgp group
Group Type: External                                Local AS: 17
  Name: external-peers  Index: 0                    Flags: <>
  Holdtime: 0
  Total peers: 4        Established: 4
  2001:db8:0:1:2a0:a502:0:1da+54987
  2001:db8:0:2:2a0:a502:0:6da+179
  2001:db8:0:3:2a0:a502:0:ada+55983
  2001:db8:0:4:2a0:a502:0:7da+49255
  inet6.0: 0/0/0/0

Groups: 1  Peers: 4    External: 4    Internal: 0    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet6.0                0          0          0          0          0          0
inet6.2                0          0          0          0          0          0
```

Meaning The group type is external, and the group has four peers.

Verifying BGP Summary Information

Purpose Verify that the BGP that the peer relationships are established.

Action From operational mode, run the **show bgp summary** command.

```
user@R1:E> show bgp summary
Groups: 1 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet6.0                0          0          0          0          0          0
inet6.2                0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
2001:db8:0:1:2a0:a502:0:1da          22       1617       1600       0       0
 12:07:00 Establ
  inet6.0: 0/0/0/0
2001:db8:0:2:2a0:a502:0:6da          22       1616       1651       0       0
 12:06:56 Establ
  inet6.0: 0/0/0/0
2001:db8:0:3:2a0:a502:0:ada          22       1617       1594       0       0
 12:04:32 Establ
  inet6.0: 0/0/0/0
2001:db8:0:4:2a0:a502:0:7da          79       1621       1599       0       0
 12:07:00 Establ
  inet6.0: 0/0/0/0
```

Meaning The Down peers: 0 output shows that the BGP peers are in the established state.

Checking the Routing Table

Purpose Verify that the inet6.0 routing table is populated with local and direct routes.

Action From operational mode, run the **show route** command.

```
user@R1:E> show route
inet6.0: 15 destinations, 18 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8::5/128    *[Direct/0] 12:41:18
                    > via lo0.5
2001:db8:0:1::/64  *[Direct/0] 14:40:01
                    > via lt-0/1/0.25
2001:db8:0:1:2a0:a502:0:19da/128
                   *[Local/0] 14:40:01
                      Local via lt-0/1/0.25
2001:db8:0:2::/64  *[Direct/0] 14:40:02
                    > via lt-0/1/0.5
2001:db8:0:2:2a0:a502:0:5da/128
                   *[Local/0] 14:40:02
                      Local via lt-0/1/0.5
2001:db8:0:3::/64  *[Direct/0] 14:40:02
                    > via lt-0/1/0.9
2001:db8:0:3:2a0:a502:0:9da/128
                   *[Local/0] 14:40:02
                      Local via lt-0/1/0.9
2001:db8:0:4::/64  *[Direct/0] 14:40:01
                    > via lt-0/1/0.21
2001:db8:0:4:2a0:a502:0:15da/128
                   *[Local/0] 14:40:01
                      Local via lt-0/1/0.21
fe80::/64          *[Direct/0] 14:40:02
                    > via lt-0/1/0.5
                    [Direct/0] 14:40:02
                    > via lt-0/1/0.9
                    [Direct/0] 14:40:01
                    > via lt-0/1/0.21
                    [Direct/0] 14:40:01
                    > via lt-0/1/0.25
fe80::2a0:a502:0:5da/128
                   *[Local/0] 14:40:02
                      Local via lt-0/1/0.5
fe80::2a0:a502:0:9da/128
                   *[Local/0] 14:40:02
                      Local via lt-0/1/0.9
fe80::2a0:a502:0:15da/128
                   *[Local/0] 14:40:01
                      Local via lt-0/1/0.21
fe80::2a0:a502:0:19da/128
                   *[Local/0] 14:40:01
                      Local via lt-0/1/0.25
fe80::2a0:a50f:fc56:1da/128
                   *[Direct/0] 12:41:18
                    > via lo0.5
```

Meaning The inet6.0 routing table contains local and direct routes. To populate the routing table with other types of routes, you must configure routing policies.

Related Documentation

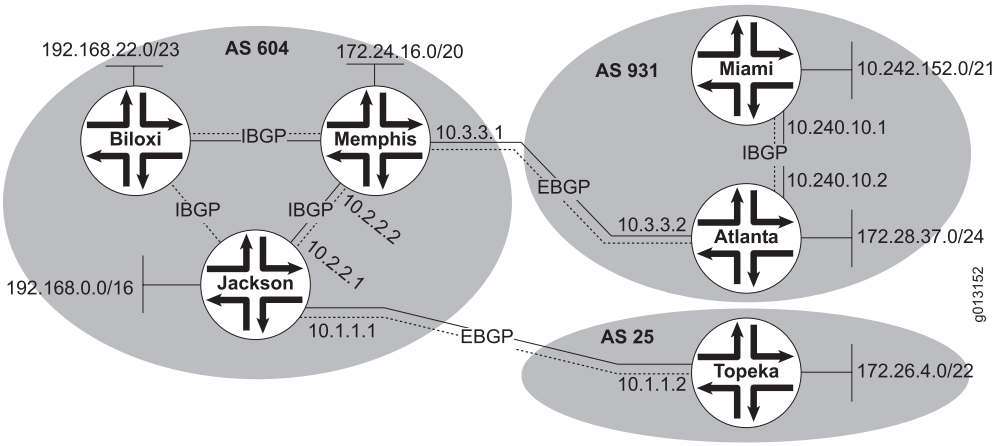
- Examples: Configuring Internal BGP Peering on page 34
- *BGP Configuration Overview*

Examples: Configuring Internal BGP Peering

Understanding Internal BGP Peering Sessions

When two BGP-enabled devices are in the same autonomous system (AS), the BGP session is called an *internal* BGP session, or IBGP session. BGP uses the same message types on IBGP and external BGP (EBGP) sessions, but the rules for when to send each message and how to interpret each message differ slightly. For this reason, some people refer to IBGP and EBGP as two separate protocols.

Figure 5: Internal and External BGP



In Figure 5 on page 34, Device Jackson, Device Memphis, and Device Biloxi have IBGP peer sessions with each other. Likewise, Device Miami and Device Atlanta have IBGP peer sessions between each other.

The purpose of IBGP is to provide a means by which EBGP route advertisements can be forwarded throughout the network. In theory, to accomplish this task you could redistribute all of your EBGP routes into an interior gateway protocol (IGP), such as OSPF or IS-IS. This, however, is not recommended in a production environment because of the large number of EBGP routes in the Internet and because of the way that IGPs operate. In short, with that many routes the IGP churns or crashes.

Generally, the loopback interface (lo0) is used to establish connections between IBGP peers. The loopback interface is always up as long as the device is operating. If there is a route to the loopback address, the IBGP peering session stays up. If a physical interface address is used instead and that interface goes up and down, the IBGP peering session also goes up and down. Thus the loopback interface provides fault tolerance in case the physical interface or the link goes down, if the device has link redundancy.

While IBGP neighbors do not need to be directly connected, they do need to be fully meshed. In this case, fully meshed means that each device is logically connected to every

other device through neighbor peer relationships. The **neighbor** statement creates the mesh. Because of the full mesh requirement of IBGP, you must configure individual peering sessions between all IBGP devices in the AS. The full mesh need not be physical links. Rather, the configuration on each routing device must create a full mesh of peer sessions (using multiple **neighbor** statements).

NOTE: **The requirement for a full mesh is waived if you configure a confederation or route reflection.**

To understand the full-mesh requirement, consider that an IBGP-learned route cannot be readvertised to another IBGP peer. The reason for preventing the readvertisement of IBGP routes and requiring the full mesh is to avoid routing loops within an AS. The AS path attribute is the means by which BGP routing devices avoid loops. The path information is examined for the local AS number only when the route is received from an EBGP peer. Because the attribute is only modified across AS boundaries, this system works well. However, the fact that the attribute is only modified across AS boundaries presents an issue inside the AS. For example, suppose that routing devices A, B, and C are all in the same AS. Device A receives a route from an EBGP peer and sends the route to Device B, which installs it as the active route. The route is then sent to Device C, which installs it locally and sends it back to Device A. If Device A installs the route, a loop is formed within the AS. The routing devices are not able to detect the loop because the AS path attribute is not modified during these advertisements. Therefore, the BGP protocol designers decided that the only assurance of never forming a routing loop was to prevent an IBGP peer from advertising an IBGP-learned route within the AS. For route reachability, the IBGP peers are fully meshed.

IBGP supports multihop connections, so IBGP neighbors can be located anywhere within the AS and often do not share a link. A recursive route lookup resolves the loopback peering address to an IP forwarding next hop. The lookup service is provided by static routes, or an IGP such as OSPF, or BGP routes.

Example: Configuring Internal BGP Peer Sessions

This example shows how to configure internal BGP peer sessions.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

In this example, you configure internal BGP (IBGP) peer sessions. The loopback interface (lo0) is used to establish connections between IBGP peers. The loopback interface is

always up as long as the device is operating. If there is a route to the loopback address, the IBGP peer session stays up. If a physical interface address is used instead and that interface goes up and down, the IBGP peer session also goes up and down. Thus, if the device has link redundancy, the loopback interface provides fault tolerance in case the physical interface or one of the links goes down.

When a device peers with a remote device's loopback interface address, the local device expects BGP update messages to come from (be sourced by) the remote device's loopback interface address. The **local-address** statement enables you to specify the source information in BGP update messages. If you omit the **local-address** statement, the expected source of BGP update messages is based on the device's source address selection rules, which normally results in the egress interface address being the expected source of update messages. When this happens, the peer session is not established because a mismatch exists between the expected source address (the egress interface of the peer) and the actual source (the loopback interface of the peer). To make sure that the expected source address matches the actual source address, specify the loopback interface address in the **local-address** statement.

Because IBGP supports multihop connections, IBGP neighbors can be located anywhere within the autonomous system (AS) and often do not share a link. A recursive route lookup resolves the loopback peer address to an IP forwarding next hop. In this example, this service is provided by OSPF. Although interior gateway protocol (IGP) neighbors do not need to be directly connected, they do need to be fully meshed. In this case, fully meshed means that each device is logically connected to every other device through neighbor peer relationships. The **neighbor** statement creates the mesh.

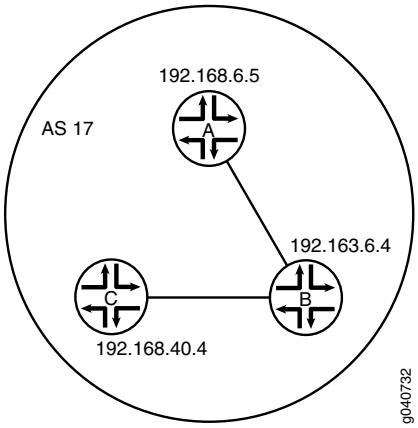
NOTE: The requirement for a full mesh is waived if you configure a confederation or route reflection.

After the BGP peers are established, BGP routes are not automatically advertised by the BGP peers. At each BGP-enabled device, policy configuration is required to export the local, static, or IGP-learned routes into the BGP routing information base (RIB) and then advertise them as BGP routes to the other peers. BGP's advertisement policy, by default, does not advertise any non-BGP routes (such as local routes) to peers.

In the sample network, the devices in AS 17 are fully meshed in the group **internal-peers**. The devices have loopback addresses 192.168.6.5, 192.163.6.4, and 192.168.40.4.

Figure 6 on page 37 shows a typical network with internal peer sessions.

Figure 6: Typical Network with IBGP Sessions



Configuration

CLI Quick Configuration

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

Device A

```
set interfaces ge-0/1/0 unit 1 description to-B
set interfaces ge-0/1/0 unit 1 family inet address 10.10.10.1/30
set interfaces lo0 unit 1 family inet address 192.168.6.5/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers description “connections to B and C”
set protocols bgp group internal-peers local-address 192.168.6.5
set protocols bgp group internal-peers export send-direct
set protocols bgp group internal-peers neighbor 192.163.6.4
set protocols bgp group internal-peers neighbor 192.168.40.4
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface ge-0/1/0.1
set policy-options policy-statement send-direct term 2 from protocol direct
set policy-options policy-statement send-direct term 2 then accept
set routing-options router-id 192.168.6.5
set routing-options autonomous-system 17
```

Device B

```
set interfaces ge-0/1/0 unit 2 description to-A
set interfaces ge-0/1/0 unit 2 family inet address 10.10.10.2/30
set interfaces ge-0/1/1 unit 5 description to-C
set interfaces ge-0/1/1 unit 5 family inet address 10.10.10.5/30
set interfaces lo0 unit 2 family inet address 192.163.6.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers description “connections to A and C”
set protocols bgp group internal-peers local-address 192.163.6.4
set protocols bgp group internal-peers export send-direct
set protocols bgp group internal-peers neighbor 192.168.40.4
set protocols bgp group internal-peers neighbor 192.168.6.5
```

```
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface ge-0/1/0.2
set protocols ospf area 0.0.0.0 interface ge-0/1/1.5
set policy-options policy-statement send-direct term 2 from protocol direct
set policy-options policy-statement send-direct term 2 then accept
set routing-options router-id 192.163.6.4
set routing-options autonomous-system 17
```

Device C

```
set interfaces ge-0/1/0 unit 6 description to-B
set interfaces ge-0/1/0 unit 6 family inet address 10.10.10.6/30
set interfaces lo0 unit 3 family inet address 192.168.40.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers description “connections to A and B”
set protocols bgp group internal-peers local-address 192.168.40.4
set protocols bgp group internal-peers export send-direct
set protocols bgp group internal-peers neighbor 192.163.6.4
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface ge-0/1/0.6
set policy-options policy-statement send-direct term 2 from protocol direct
set policy-options policy-statement send-direct term 2 then accept
set routing-options router-id 192.168.40.4
set routing-options autonomous-system 17
```

Configuring Device A

Step-by-Step Procedure

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

To configure internal BGP peer sessions on Device A:

1. Configure the interfaces.

```
[edit interfaces ge-0/1/0 unit 1]
user@A# set description to-B
user@A# set family inet address 10.10.10.1/30

[edit interfaces]
user@A# set lo0 unit 1 family inet address 192.168.6.5/32
```

2. Configure BGP.

   The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```
[edit protocols bgp group internal-peers]
user@A# set type internal
user@A# set description “connections to B and C”
user@A# set local-address 192.168.6.5
user@A# set export send-direct
user@A# set neighbor 192.163.6.4
user@A# set neighbor 192.168.40.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
```

```
user@A# set interface lo0.1 passive
user@A# set interface ge-0/1/0.1
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 2]
user@A# set from protocol direct
user@A# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@A# set router-id 192.168.6.5
user@A# set autonomous-system 17
```

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

```
user@A# show interfaces
ge-0/1/0 {
  unit 1 {
    description to-B;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.6.5/32;
    }
  }
}
```

```
user@A# show policy-options
policy-statement send-direct {
  term 2 {
    from protocol direct;
    then accept;
  }
}
```

```
user@A# show protocols
bgp {
  group internal-peers {
    type internal;
    description “connections to B and C”;
    local-address 192.168.6.5;
    export send-direct;
    neighbor 192.163.6.4;
    neighbor 192.168.40.4;
```

```
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.1 {
        passive;
      }
      interface ge-0/1/0.1;
    }
  }
```

```
user@A# show routing-options
router-id 192.168.6.5;
autonomous-system 17;
```

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

Configuring Device B

Step-by-Step Procedure

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

To configure internal BGP peer sessions on Device B:

1. Configure the interfaces.

```
[edit interfaces ge-0/1/0 unit 2]
user@B# set description to-A
user@B# set family inet address 10.10.10.2/30

[edit interfaces ge-0/1/1]
user@B# set unit 5 description to-C
user@B# set unit 5 family inet address 10.10.10.5/30

[edit interfaces]
user@B# set lo0 unit 2 family inet address 192.163.6.4/32
```

2. Configure BGP.

   The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```
[edit protocols bgp group internal-peers]
user@B# set type internal
user@B# set description "connections to A and C"
user@B# set local-address 192.163.6.4
user@B# set export send-direct
user@B# set neighbor 192.168.40.4
user@B# set neighbor 192.168.6.5
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@B# set interface lo0.2 passive
user@B# set interface ge-0/1/0.2
user@B# set interface ge-0/1/1.5
```


4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

   ```
   [edit policy-options policy-statement send-direct term 2]
   user@B# set from protocol direct
   user@B# set then accept
   ```

5. Configure the router ID and the AS number.

   ```
   [edit routing-options]
   user@B# set router-id 192.163.6.4
   user@B# set autonomous-system 17
   ```

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

```
user@B# show interfaces
ge-0/1/0 {
    unit 2 {
        description to-A;
        family inet {
            address 10.10.10.2/30;
        }
    }
}
ge-0/1/1 {
    unit 5 {
        description to-C;
        family inet {
            address 10.10.10.5/30;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 192.163.6.4/32;
        }
    }
}
```

```
user@B# show policy-options
policy-statement send-direct {
    term 2 {
        from protocol direct;
        then accept;
    }
}
```

```
user@B# show protocols
bgp {
    group internal-peers {
        type internal;
```

```
    description “connections to A and C”;
    local-address 192.163.6.4;
    export send-direct;
    neighbor 192.168.40.4;
    neighbor 192.168.6.5;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.2 {
      passive;
    }
    interface ge-0/1/0.2;
    interface ge-0/1/1.5;
  }
}
```

```
user@B# show routing-options
router-id 192.163.6.4;
autonomous-system 17;
```

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

Configuring Device C

Step-by-Step Procedure

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

To configure internal BGP peer sessions on Device C:

1. Configure the interfaces.

```
[edit interfaces ge-0/1/0 unit 6]
user@C# set description to-B
user@C# set family inet address 10.10.10.6/30
```

```
[edit interfaces]
user@C# set lo0 unit 3 family inet address 192.168.40.4/32
```

2. Configure BGP.

   The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```
[edit protocols bgp group internal-peers]
user@C# set type internal
user@C# set description “connections to A and B”
user@C# set local-address 192.168.40.4
user@C# set export send-direct
user@C# set neighbor 192.163.6.4
user@C# set neighbor 192.168.6.5
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@C# set interface lo0.3 passive
user@C# set interface ge-0/1/0.6
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

   ```
   [edit policy-options policy-statement send-direct term 2]
   user@C# set from protocol direct
   user@C# set then accept
   ```

5. Configure the router ID and the AS number.

   ```
   [edit routing-options]
   user@C# set router-id 192.168.40.4
   user@C# set autonomous-system 17
   ```

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

```
user@C# show interfaces
ge-0/1/0 {
  unit 6 {
    description to-B;
    family inet {
      address 10.10.10.6/30;
    }
  }
}
lo0 {
  unit 3 {
    family inet {
      address 192.168.40.4/32;
    }
  }
}
```

```
user@C# show policy-options
policy-statement send-direct {
  term 2 {
    from protocol direct;
    then accept;
  }
}
```

```
user@C# show protocols
bgp {
  group internal-peers {
    type internal;
    description “connections to A and B”;
    local-address 192.168.40.4;
    export send-direct;
    neighbor 192.163.6.4;
    neighbor 192.168.6.5;
  }
}
ospf {
```

```
    area 0.0.0.0 {
      interface lo0.3 {
        passive;
      }
      interface ge-0/1/0.6;
    }
  }
```

```
user@C# show routing-options
router-id 192.168.40.4;
autonomous-system 17;
```

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

Verification

Confirm that the configuration is working properly.

- Verifying BGP Neighbors on page 44
- Verifying BGP Groups on page 45
- Verifying BGP Summary Information on page 46
- Verifying That BGP Routes Are Installed in the Routing Table on page 46

Verifying BGP Neighbors

Purpose Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

Action From operational mode, enter the **show bgp neighbor** command.

```
user@A> show bgp neighbor
Peer: 192.163.6.4+179 AS 17    Local: 192.168.6.5+58852 AS 17
  Type: Internal    State: Established    Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-direct ]
  Options: Preference LocalAddress Refresh
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.163.6.4     Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
```

```
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            3
    Accepted prefixes:            3
    Suppressed due to damping:    0
    Advertised prefixes:          2
  Last traffic (seconds): Received 25   Sent 19   Checked 67
  Input messages:  Total 2420   Updates 4       Refreshes 0     Octets 46055
  Output messages: Total 2411   Updates 2       Refreshes 0     Octets 45921
  Output Queue[0]: 0

Peer: 192.168.40.4+179 AS 17   Local: 192.168.6.5+56466 AS 17
  Type: Internal    State: Established    Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-direct ]
  Options: Preference LocalAddress Refresh
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.40.4    Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            2
    Accepted prefixes:            2
    Suppressed due to damping:    0
    Advertised prefixes:          2
  Last traffic (seconds): Received 7    Sent 21   Checked 24
  Input messages:  Total 2412   Updates 2       Refreshes 0     Octets 45867
  Output messages: Total 2409   Updates 2       Refreshes 0     Octets 45883
  Output Queue[0]: 0
```

Verifying BGP Groups

Purpose Verify that the BGP groups are configured correctly.

Action From operational mode, enter the **show bgp group** command.

```
user@A> show bgp group
Group Type: Internal    AS: 17                     Local AS: 17
  Name: internal-peers  Index: 0                   Flags: <Export Eval>
  Export: [ send-direct ]
  Holdtime: 0
  Total peers: 2        Established: 2
  192.163.6.4+179
```

```
  192.168.40.4+179
  inet.0: 0/5/5/0

Groups: 1  Peers: 2    External: 0    Internal: 2    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 5          0          0          0          0          0
```

Verifying BGP Summary Information

Purpose Verify that the BGP configuration is correct.

Action From operational mode, enter the **show bgp summary** command.

```
user@A> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 5          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.163.6.4              17       2441       2432       0       0    18:18:52
0/3/3/0             0/0/0/0
192.168.40.4             17       2432       2430       0       0    18:18:48
0/2/2/0             0/0/0/0
```

Verifying That BGP Routes Are Installed in the Routing Table

Purpose Verify that the export policy configuration is causing the BGP routes to be installed in the routing tables of the peers.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@A> show route protocol bgp
inet.0: 7 destinations, 12 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30       [BGP/170] 07:09:57, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via ge-0/1/0.1
10.10.10.4/30       [BGP/170] 07:09:57, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via ge-0/1/0.1
                    [BGP/170] 07:07:12, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via ge-0/1/0.1
192.163.6.4/32      [BGP/170] 07:09:57, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via ge-0/1/0.1
192.168.40.4/32     [BGP/170] 07:07:12, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via ge-0/1/0.1
```

Example: Configuring Internal BGP Peering Sessions on Logical Systems

This example shows how to configure internal BGP peer sessions on logical systems.

Requirements

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

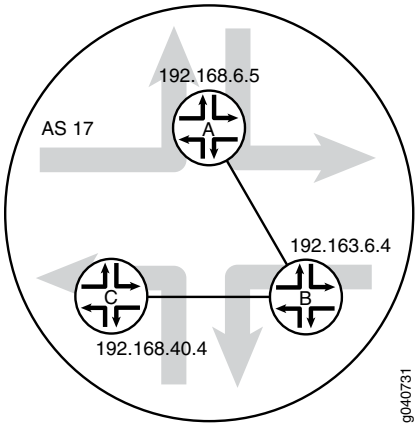
Overview

In this example, you configure internal BGP (IBGP) peering sessions.

In the sample network, the devices in AS 17 are fully meshed in the group **internal-peers**. The devices have loopback addresses 192.168.6.5, 192.163.6.4, and 192.168.40.4.

Figure 7 on page 47 shows a typical network with internal peer sessions.

Figure 7: Typical Network with IBGP Sessions



Configuration

CLI Quick Configuration

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

```
set logical-systems A interfaces lt-0/1/0 unit 1 description to-B
set logical-systems A interfaces lt-0/1/0 unit 1 encapsulation ethernet
set logical-systems A interfaces lt-0/1/0 unit 1 peer-unit 2
set logical-systems A interfaces lt-0/1/0 unit 1 family inet address 10.10.10.1/30
set logical-systems A interfaces lo0 unit 1 family inet address 192.168.6.5/32
set logical-systems A protocols bgp group internal-peers type internal
set logical-systems A protocols bgp group internal-peers local-address 192.168.6.5
set logical-systems A protocols bgp group internal-peers export send-direct
set logical-systems A protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems A protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems A protocols ospf area 0.0.0.0 interface lo0.1 passive
set logical-systems A protocols ospf area 0.0.0.0 interface lt-0/1/0.1
set logical-systems A policy-options policy-statement send-direct term 2 from protocol
    direct
set logical-systems A policy-options policy-statement send-direct term 2 then accept
```

```
set logical-systems A routing-options router-id 192.168.6.5
set logical-systems A routing-options autonomous-system 17
set logical-systems B interfaces lt-0/1/0 unit 2 description to-A
set logical-systems B interfaces lt-0/1/0 unit 2 encapsulation ethernet
set logical-systems B interfaces lt-0/1/0 unit 2 peer-unit 1
set logical-systems B interfaces lt-0/1/0 unit 2 family inet address 10.10.10.2/30
set logical-systems B interfaces lt-0/1/0 unit 5 description to-C
set logical-systems B interfaces lt-0/1/0 unit 5 encapsulation ethernet
set logical-systems B interfaces lt-0/1/0 unit 5 peer-unit 6
set logical-systems B interfaces lt-0/1/0 unit 5 family inet address 10.10.10.5/30
set logical-systems B interfaces lo0 unit 2 family inet address 192.163.6.4/32
set logical-systems B protocols bgp group internal-peers type internal
set logical-systems B protocols bgp group internal-peers local-address 192.163.6.4
set logical-systems B protocols bgp group internal-peers export send-direct
set logical-systems B protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems B protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems B protocols ospf area 0.0.0.0 interface lo0.2 passive
set logical-systems B protocols ospf area 0.0.0.0 interface lt-0/1/0.2
set logical-systems B protocols ospf area 0.0.0.0 interface lt-0/1/0.5
set logical-systems B policy-options policy-statement send-direct term 2 from protocol
    direct
set logical-systems B policy-options policy-statement send-direct term 2 then accept
set logical-systems B routing-options router-id 192.163.6.4
set logical-systems B routing-options autonomous-system 17
set logical-systems C interfaces lt-0/1/0 unit 6 description to-B
set logical-systems C interfaces lt-0/1/0 unit 6 encapsulation ethernet
set logical-systems C interfaces lt-0/1/0 unit 6 peer-unit 5
set logical-systems C interfaces lt-0/1/0 unit 6 family inet address 10.10.10.6/30
set logical-systems C interfaces lo0 unit 3 family inet address 192.168.40.4/32
set logical-systems C protocols bgp group internal-peers type internal
set logical-systems C protocols bgp group internal-peers local-address 192.168.40.4
set logical-systems C protocols bgp group internal-peers export send-direct
set logical-systems C protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems C protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems C protocols ospf area 0.0.0.0 interface lo0.3 passive
set logical-systems C protocols ospf area 0.0.0.0 interface lt-0/1/0.6
set logical-systems C policy-options policy-statement send-direct term 2 from protocol
    direct
set logical-systems C policy-options policy-statement send-direct term 2 then accept
set logical-systems C routing-options router-id 192.168.40.4
set logical-systems C routing-options autonomous-system 17
```

Device A

Step-by-Step Procedure

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

To configure internal BGP peer sessions on Device A:

1. Configure the interfaces.

```
[edit logical-systems A interfaces lt-0/1/0 unit 1]
user@R1# set description to-B
user@R1# set encapsulation ethernet
user@R1# set peer-unit 2
```

```
user@R1# set family inet address 10.10.10.1/30
user@R1# set family inet address 192.168.6.5/32
user@R1# up
user@R1# up
[edit logical-systems A interfaces]
user@R1# set lo0 unit 1 family inet address 192.168.6.5/32
user@R1# exit
[edit]
user@R1# edit logical-systems B interfaces lt-0/1/0
[edit logical-systems B interfaces lt-0/1/0]
user@R1# set unit 2 description to-A
user@R1# set unit 2 encapsulation ethernet
user@R1# set unit 2 peer-unit 1
user@R1# set unit 2 family inet address 10.10.10.2/30
user@R1# set unit 5 description to-C
user@R1# set unit 5 encapsulation ethernet
user@R1# set unit 5 peer-unit 6
user@R1# set family inet address 10.10.10.5/30
user@R1# up
[edit logical-systems B interfaces]
user@R1# set lo0 unit 2 family inet address 192.163.6.4/32
user@R1# exit
[edit]
user@R1# edit logical-systems C interfaces lt-0/1/0 unit 6
[edit logical-systems C interfaces lt-0/1/0 unit 6]
set description to-B
set encapsulation ethernet
set peer-unit 5
set family inet address 10.10.10.6/30
user@R1# up
user@R1# up
[edit logical-systems C interfaces]
set lo0 unit 3 family inet address 192.168.40.4/32
```

2. Configure BGP.

   On Logical System A, the **neighbor** statements are included for both Device B and Device C, even though Logical System A is not directly connected to Device C.

```
[edit logical-systems A protocols bgp group internal-peers]
user@R1# set type internal
user@R1# set local-address 192.168.6.5
user@R1# set export send-direct
user@R1# set neighbor 192.163.6.4
user@R1# set neighbor 192.168.40.4

[edit logical-systems B protocols bgp group internal-peers]
user@R1# set type internal
user@R1# set local-address 192.163.6.4
user@R1# set export send-direct
user@R1# set neighbor 192.168.40.4
user@R1# set neighbor 192.168.6.5

[edit logical-systems C protocols bgp group internal-peers]
user@R1# set type internal
user@R1# set local-address 192.168.40.4
```

```
user@R1# set export send-direct
user@R1# set neighbor 192.163.6.4
user@R1# set neighbor 192.168.6.5
```

3. Configure OSPF.

```
[edit logical-systems A protocols ospf area 0.0.0.0]
user@R1# set interface lo0.1 passive
user@R1# set interface lt-0/1/0.1

[edit logical-systems A protocols ospf area 0.0.0.0]
user@R1# set interface lo0.2 passive
user@R1# set interface lt-0/1/0.2
user@R1# set interface lt-0/1/0.5

[edit logical-systems A protocols ospf area 0.0.0.0]
user@R1# set interface lo0.3 passive
user@R1# set interface lt-0/1/0.6
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit logical-systems A policy-options policy-statement send-direct term 2]
user@R1# set from protocol direct
user@R1# set then accept

[edit logical-systems B policy-options policy-statement send-direct term 2]
user@R1# set from protocol direct
user@R1# set then accept

[edit logical-systems C policy-options policy-statement send-direct term 2]
user@R1# set from protocol direct
user@R1# set then accept
```

5. Configure the router ID and the autonomous system (AS) number.

```
[edit logical-systems A routing-options]
user@R1# set router-id 192.168.6.5
user@R1# set autonomous-system 17

[edit logical-systems B routing-options]
user@R1# set router-id 192.163.6.4
user@R1# set autonomous-system 17

[edit logical-systems C routing-options]
user@R1# set router-id 192.168.40.4
user@R1# set autonomous-system 17
```

Results From configuration mode, confirm your configuration by entering the **show logical-systems** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show logical-systems
```



```
A {
    interfaces {
        lt-0/1/0 {
            unit 1 {
                description to-B;
                encapsulation ethernet;
                peer-unit 2;
                family inet {
                    address 10.10.10.1/30;
                }
            }
        }
        lo0 {
            unit 1 {
                family inet {
                    address 192.168.6.5/32;
                }
            }
        }
    }
    protocols {
        bgp {
            group internal-peers {
                type internal;
                local-address 192.168.6.5;
                export send-direct;
                neighbor 192.163.6.4;
                neighbor 192.168.40.4;
            }
        }
        ospf {
            area 0.0.0.0 {
                interface lo0.1 {
                    passive;
                }
                interface lt-0/1/0.1;
            }
        }
    }
    policy-options {
        policy-statement send-direct {
            term 2 {
                from protocol direct;
                then accept;
            }
        }
    }
    routing-options {
        router-id 192.168.6.5;
        autonomous-system 17;
    }
}
B {
    interfaces {
        lt-0/1/0 {
            unit 2 {
```

```
        description to-A;
        encapsulation ethernet;
        peer-unit 1;
        family inet {
          address 10.10.10.2/30;
        }
      }
      unit 5 {
        description to-C;
        encapsulation ethernet;
        peer-unit 6;
        family inet {
          address 10.10.10.5/30;
        }
      }
    }
    lo0 {
      unit 2 {
        family inet {
          address 192.163.6.4/32;
        }
      }
    }
}
protocols {
    bgp {
      group internal-peers {
        type internal;
        local-address 192.163.6.4;
        export send-direct;
        neighbor 192.168.40.4;
        neighbor 192.168.6.5;
      }
    }
    ospf {
      area 0.0.0.0 {
        interface lo0.2 {
          passive;
        }
        interface lt-0/1/0.2;
        interface lt-0/1/0.5;
      }
    }
}
policy-options {
    policy-statement send-direct {
      term 2 {
        from protocol direct;
        then accept;
      }
    }
}
routing-options {
    router-id 192.163.6.4;
    autonomous-system 17;
}
```

```
}
C {
    interfaces {
        lt-0/1/0 {
            unit 6 {
                description to-B;
                encapsulation ethernet;
                peer-unit 5;
                family inet {
                    address 10.10.10.6/30;
                }
            }
        }
        lo0 {
            unit 3 {
                family inet {
                    address 192.168.40.4/32;
                }
            }
        }
    }
    protocols {
        bgp {
            group internal-peers {
                type internal;
                local-address 192.168.40.4;
                export send-direct;
                neighbor 192.163.6.4;
                neighbor 192.168.6.5;
            }
        }
        ospf {
            area 0.0.0.0 {
                interface lo0.3 {
                    passive;
                }
                interface lt-0/1/0.6;
            }
        }
    }
    policy-options {
        policy-statement send-direct {
            term 2 {
                from protocol direct;
                then accept;
            }
        }
    }
    routing-options {
        router-id 192.168.40.4;
        autonomous-system 17;
    }
}
```

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

Verification

Confirm that the configuration is working properly.

Verifying BGP Neighbors

Purpose Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

Action From the operational mode, enter the **show bgp neighbor** command.

```
user@R1> show bgp neighbor logical-system A
Peer: 192.163.6.4+179 AS 17    Local: 192.168.6.5+58852 AS 17
  Type: Internal    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-direct ]
  Options: <Preference LocalAddress Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.163.6.4     Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            3
    Accepted prefixes:            3
    Suppressed due to damping:    0
    Advertised prefixes:          2
  Last traffic (seconds): Received 16   Sent 1    Checked 63
  Input messages:  Total 15713  Updates 4       Refreshes 0     Octets 298622
  Output messages: Total 15690  Updates 2       Refreshes 0     Octets 298222
  Output Queue[0]: 0

Peer: 192.168.40.4+179 AS 17   Local: 192.168.6.5+56466 AS 17
  Type: Internal    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
```

```
  Export: [ send-direct ]
  Options: <Preference LocalAddress Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.40.4    Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            2
    Accepted prefixes:            2
    Suppressed due to damping:    0
    Advertised prefixes:          2
  Last traffic (seconds): Received 15   Sent 22   Checked 68
  Input messages:  Total 15688  Updates 2       Refreshes 0     Octets 298111
  Output messages: Total 15688  Updates 2       Refreshes 0     Octets 298184
  Output Queue[0]: 0
```

Verifying BGP Groups

Purpose Verify that the BGP groups are configured correctly.

Action From the operational mode, enter the **show bgp group** command.

```
user@A> show bgp group logical-system A
Group Type: Internal    AS: 17                     Local AS: 17
  Name: internal-peers  Index: 0                   Flags: <Export Eval>
  Export: [ send-direct ]
  Holdtime: 0
  Total peers: 2        Established: 2
  192.163.6.4+179
  192.168.40.4+179
  inet.0: 0/5/5/0

Groups: 1  Peers: 2    External: 0    Internal: 2    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 5          0          0          0          0          0
```

Verifying BGP Summary Information

Purpose Verify that the BGP configuration is correct.

Action From the operational mode, enter the **show bgp summary** command.

```
user@A> show bgp summary logical-system A
```

```
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 5          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.163.6.4              17      15723      15700       0       0 4d 22:13:15
0/3/3/0             0/0/0/0
192.168.40.4             17      15698      15699       0       0 4d 22:13:11
0/2/2/0             0/0/0/0
```

Verifying That BGP Routes Are Installed in the Routing Table

Purpose Verify that the export policy configuration is working.

Action From the operational mode, enter the **show route protocol bgp** command.

```
user@A> show route protocol bgp logical-system A
inet.0: 7 destinations, 12 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30       [BGP/170] 4d 11:05:55, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via lt-0/1/0.1
10.10.10.4/30       [BGP/170] 4d 11:05:55, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via lt-0/1/0.1
                    [BGP/170] 4d 11:03:10, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via lt-0/1/0.1
192.163.6.4/32      [BGP/170] 4d 11:05:55, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via lt-0/1/0.1
192.168.40.4/32     [BGP/170] 4d 11:03:10, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via lt-0/1/0.1
```

Related Documentation

- Examples: Configuring External BGP Peering on page 11

CHAPTER 3

BGP Path Attribute Configuration

Example: Configuring BGP Local Preference

Understanding the BGP Local Preference

Internal BGP (IBGP) sessions use a metric called the *local preference*, which is carried in IBGP update packets in the path attribute LOCAL_PREF. When an autonomous system (AS) has multiple routes to another AS, the local preference indicates the degree of preference for one route over the other routes. The route with the highest local preference value is preferred.

The LOCAL_PREF path attribute is always advertised to IBGP peers and to neighboring confederations. It is never advertised to external BGP (EBGP) peers. The default behavior is to not modify the LOCAL_PREF path attribute if it is present.

The LOCAL_PREF path attribute applies at export time only, when the routes are exported from the routing table into BGP.

If a BGP route is received without a LOCAL_PREF attribute, the route is stored in the routing table and advertised by BGP as if it were received with a LOCAL_PREF value of 100. A non-BGP route that is advertised by BGP is advertised with a LOCAL_PREF value of 100 by default.

Example: Configuring the Local Preference Value for BGP Routes

This example shows how to configure local preference in internal BGP (IBGP) peer sessions.

Requirements

No special configuration beyond device initialization is required before you configure this example.

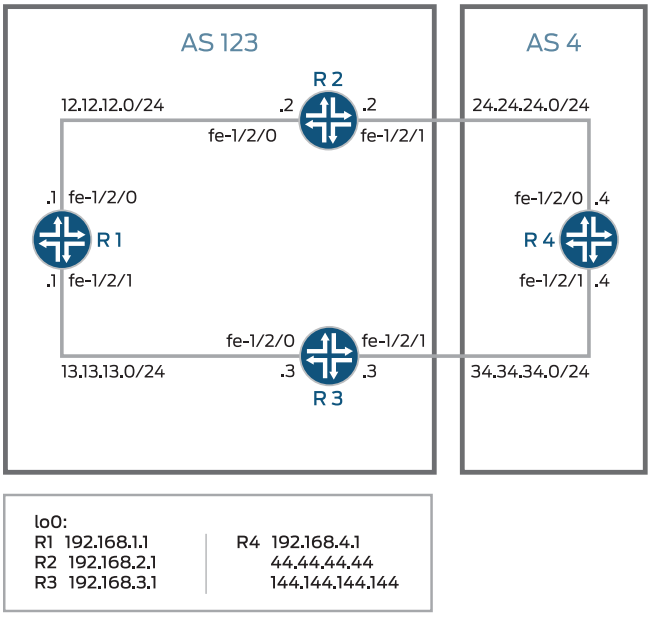
Overview

To change the local preference metric advertised in the path attribute, you must include the local-preference statement, specifying a value from 0 through 4,294,967,295 ($2^{32}-1$).

There are several reasons you might want to prefer one path over another. For example, compared to other paths, one path might be less expensive to use, might have higher bandwidth, or might be more stable.

Figure 8 on page 58 shows a typical network with internal peer sessions and multiple exit points to a neighboring AS.

Figure 8: Typical Network with IBGP Sessions and Multiple Exit Points



To reach Device R4, Device R1 can take a path through either Device R2 or Device R3. By default, the local preference is 100 for either route. When the local preferences are equal, Junos OS has rules for breaking the tie and choosing a path. (See "Understanding BGP Path Selection" on page 197.) In this example, the active route is through Device R2 because the router ID of Device R2 is lower than the router ID of Device R3. The following example shows how to override the default behavior with an explicit setting for the local preference. The example configures a local preference of 300 on Device R3, thereby making Device R3 the preferred path to reach Device R4.

Configuration

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 12.12.12.1/24
set interfaces fe-1/2/1 unit 2 family inet address 13.13.13.1/24
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.1.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
set protocols ospf area 0.0.0.0 interface fe-1/2/1.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.1.1
```

Device R2

```
set interfaces fe-1/2/0 unit 3 family inet address 12.12.12.2/24
set interfaces fe-1/2/1 unit 4 family inet address 24.24.24.2/24
set interfaces lo0 unit 2 family inet address 192.168.2.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.2.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 24.24.24.4
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.2.1
```

Device R3

```
set interfaces fe-1/2/0 unit 5 family inet address 13.13.13.3/24
set interfaces fe-1/2/1 unit 6 family inet address 34.34.34.3/24
set interfaces lo0 unit 3 family inet address 192.168.3.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.3.1
```

```
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 34.34.34.4
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.5
set protocols ospf area 0.0.0.0 interface fe-1/2/1.6
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.3.1
```

Device R4

```
set interfaces fe-1/2/0 unit 7 family inet address 24.24.24.4/24
set interfaces fe-1/2/1 unit 8 family inet address 34.34.34.4/24
set interfaces lo0 unit 4 family inet address 192.168.4.1/32
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 123
set protocols bgp group external neighbor 34.34.34.3
set protocols bgp group external neighbor 24.24.24.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 4
set routing-options router-id 192.168.4.1
```

Configuring Device R1

Step-by-Step Procedure

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

To configure Device R1:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 1]
user@R1# set family inet address 12.12.12.1/24

[edit interfaces fe-1/2/1 unit 2]
user@R1# set family inet address 13.13.13.1/24

[edit interfaces lo0 unit 1]
user@R1# set family inet address 192.168.1.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R1# set type internal
user@R1# set local-address 192.168.1.1
user@R1# set export send-direct
user@R1# set neighbor 192.168.2.1
user@R1# set neighbor 192.168.3.1
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface lo0.1 passive
user@R1# set interface fe-1/2/0.1
user@R1# set interface fe-1/2/1.2
```

4. Configure a policy that accepts direct routes.

NOTE: Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R1# set from protocol direct
user@R1# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 123
user@R1# set router-id 192.168.1.1
```

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

```
user@R1# show interfaces
fe-1/2/0 {
   unit 1 {
      family inet {
         address 12.12.12.1/24;
      }
   }
}
fe-1/2/1 {
   unit 2 {
      family inet {
         address 13.13.13.1/24;
      }
   }
}
lo0 {
   unit 1 {
      family inet {
         address 192.168.1.1/32;
      }
   }
}
```

```
user@R1# show policy-options
policy-statement send-direct {
   term 1 {
      from protocol direct;
```

```
        then accept;
    }
}
```

```
user@R1# show protocols
bgp {
    group internal {
        type internal;
        local-address 192.168.1.1;
        export send-direct;
        neighbor 192.168.2.1;
        neighbor 192.168.3.1;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.1 {
            passive;
        }
        interface fe-1/2/0.1;
        interface fe-1/2/1.2;
    }
}
```

```
user@R1# show routing-options
autonomous-system 123;
router-id 192.168.1.1;
```

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

Configuring Device R2

Step-by-Step Procedure

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

To configure Device R2:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 3]
user@R2# set family inet address 12.12.12.21/24

[edit interfaces fe-1/2/1 unit 4]
user@R2# set family inet address 24.24.24.2/24

[edit interfaces lo0 unit 2]
user@R2# set family inet address 192.168.2.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R2# set type internal
user@R2# set local-address 192.168.2.1
user@R2# set export send-direct
user@R2# set neighbor 192.168.1.1
user@R2# set neighbor 192.168.3.1
```

```
[edit protocols bgp group external]
user@R2# set type external
user@R2# set export send-direct
user@R2# set peer-as 4
user@R2# set neighbor 24.24.24.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R2# set interface lo0.2 passive
user@R2# set interface fe-1/2/0.3
user@R2# set interface fe-1/2/1.4
```

4. Configure a policy that accepts direct routes.

NOTE: Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R2# set from protocol direct
user@R2# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R2# set autonomous-system 123
user@R2# set router-id 192.168.2.1
```

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

```
user@R2# show interfaces
fe-1/2/0 {
    unit 3 {
        family inet {
            address 12.12.12.2/24;
        }
    }
}
fe-1/2/1 {
    unit 4 {
        family inet {
            address 24.24.24.2/24;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 192.168.2.1/32;
        }
    }
```

```
}

user@R2# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}

user@R2# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.2.1;
    export send-direct;
    neighbor 192.168.1.1;
    neighbor 192.168.3.1;
  }
  group external {
    type external;
    export send-direct;
    peer-as 4;
    neighbor 24.24.24.4;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.2 {
      passive;
    }
    interface fe-1/2/0.3;
    interface fe-1/2/1.4;
  }
}

user@R2# show routing-options
autonomous-system 123;
router-id 192.168.2.1;
```

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

Configuring Device R3

Step-by-Step Procedure

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

To configure Device R3:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 5]
user@R3# set family inet address 13.13.13.3/24

[edit interfaces fe-1/2/1 unit 6]
user@R3# set family inet address 34.34.34.3/24
```

```
[edit interfaces lo0 unit 3]
user@R3# set family inet address 192.168.3.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R3# set type internal
user@R3# set local-address 192.168.3.1
user@R3# set export send-direct
user@R3# set neighbor 192.168.1.1
user@R3# set neighbor 192.168.2.1

[edit protocols bgp group external]
user@R3# set type external
user@R3# set export send-direct
user@R3# set peer-as 4
user@R3# set neighbor 34.34.34.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R3# set interface lo0.3 passive
user@R3# set interface fe-1/2/0.5
user@R3# set interface fe-1/2/1.6
```

4. Configure a policy that accepts direct routes.

NOTE: Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R3# set from protocol direct
user@R3# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R3# set autonomous-system 123
user@R3# set router-id 192.168.3.1
```

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

```
user@R3# show interfaces
fe-1/2/0 {
   unit 5 {
      family inet {
         address 13.13.13.3/24;
      }
   }
}
fe-1/2/1 {
```

```
    unit 6 {
      family inet {
        address 34.34.34.3/24;
      }
    }
  }
  lo0 {
    unit 3 {
      family inet {
        address 192.168.3.1/32;
      }
    }
  }
```

```
user@R3# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
```

```
user@R3# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.3.1;
    export send-direct;
    neighbor 192.168.1.1;
    neighbor 192.168.2.1;
  }
  group external {
    type external;
    export send-direct;
    peer-as 4;
    neighbor 34.34.34.4;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.3 {
      passive;
    }
    interface fe-1/2/0.5;
    interface fe-1/2/1.6;
  }
}
```

```
user@R3# show routing-options
autonomous-system 123;
router-id 192.168.3.1;
```

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

Configuring Device R4

Step-by-Step Procedure

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

To configure Device R4:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 7]
user@R4# set family inet address 24.24.24.4/24

[edit interfaces fe-1/2/1 unit 8]
user@R4# set family inet address 34.34.34.4/24

[edit interfaces lo0 unit 4]
user@R4# set family inet address 192.168.4.1/32
```

2. Configure BGP.

```
[edit protocols bgp group external]
user@R4# set type external
user@R4# set export send-direct
user@R4# set peer-as 123
user@R4# set neighbor 34.34.34.3
user@R4# set neighbor 24.24.24.2
```

3. Configure a policy that accepts direct routes.

NOTE: Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R4# set from protocol direct
user@R4# set then accept
```

4. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R4# set autonomous-system 4
user@R4# set router-id 192.168.4.1
```

Results

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

```
user@R4# show interfaces
fe-1/2/0 {
  unit 7 {
    family inet {
      address 24.24.24.4/24;
```

```
    }
  }
}
fe-1/2/1 {
  unit 8 {
    family inet {
      address 34.34.34.4/24;
    }
  }
}
lo0 {
  unit 4 {
    family inet {
      address 192.168.4.1/32;
    }
  }
}
```

```
user@R4# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
```

```
user@R4# show protocols
bgp {
  group external {
    type external;
    export send-direct;
    peer-as 123;
    neighbor 34.34.34.3;
    neighbor 24.24.24.2;
  }
}
```

```
user@R4# show routing-options
autonomous-system 4;
router-id 192.168.4.1;
```

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

Verification

Confirm that the configuration is working properly.

- Checking the Active Path From Device R1 to Device R4 on page 68
- Altering the Local Preference to Change the Path Selection on page 69
- Rechecking the Active Path From Device R1 to Device R4 on page 69

Checking the Active Path From Device R1 to Device R4

Purpose Verify that the active path from Device R1 to Device R4 goes through Device R2.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@R1> show route protocol bgp
inet.0: 11 destinations, 18 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

12.12.12.0/24       [BGP/170] 00:11:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
13.13.13.0/24       [BGP/170] 00:11:48, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
24.24.24.0/24       [BGP/170] 00:11:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
34.34.34.0/24       [BGP/170] 00:11:48, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.2.1/32      [BGP/170] 00:11:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
192.168.3.1/32      [BGP/170] 00:11:48, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.4.1/32     *[BGP/170] 00:05:14, localpref 100, from 192.168.2.1
                      AS path: 4 I
                    > to 12.12.12.2 via fe-1/2/0.1
                    [BGP/170] 00:05:14, localpref 100, from 192.168.3.1
                      AS path: 4 I
                    > to 13.13.13.3 via fe-1/2/1.2
```

Meaning The asterisk (*) shows that the preferred path is through Device R2. In the default configuration, Device R2 has a lower router ID than Device R3. The router ID is controlling the path selection.

Altering the Local Preference to Change the Path Selection

Purpose Change the path so that it goes through Device R3.

Action From configuration mode, enter the **set local-preference 300** command.

```
[edit protocols bgp group internal]
user@R3# set local-preference 300
user@R3# commit
```

Rechecking the Active Path From Device R1 to Device R4

Purpose Verify that the active path from Device R1 to Device R4 goes through Device R3.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@R1> show route protocol bgp
inet.0: 11 destinations, 17 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

12.12.12.0/24       [BGP/170] 00:16:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
13.13.13.0/24       [BGP/170] 00:00:22, localpref 300, from 192.168.3.1
```

```
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
24.24.24.0/24       [BGP/170] 00:16:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
34.34.34.0/24       [BGP/170] 00:00:22, localpref 300, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.2.1/32      [BGP/170] 00:16:48, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
192.168.3.1/32      [BGP/170] 00:00:22, localpref 300, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.4.1/32     *[BGP/170] 00:00:21, localpref 300, from 192.168.3.1
                      AS path: 4 I
                    > to 13.13.13.3 via fe-1/2/1.2
```

Meaning The asterisk (*) shows that the preferred path is through Device R3. In the altered configuration, Device R3 has a higher local preference than Device R2. The local preference is controlling the path selection.

Related Documentation

- Examples: Configuring Internal BGP Peering on page 34
- *BGP Configuration Overview*

Examples: Configuring BGP MED

Understanding the MED Attribute

The BGP multiple exit discriminator (MED, or MULTI_EXIT_DISC) is a non-transitive attribute, meaning that it is not propagated throughout the Internet, but only to adjacent autonomous systems (ASs). The MED attribute is optional, meaning that it is not always sent with the BGP updates. The purpose of MED is to influence how other ASs enter your AS to reach a certain prefix.

The MED attribute has a value that is referred to as a *metric*. If all other factors in determining an exit point are equal, the exit point with the lowest metric is preferred.

If a MED is received over an external BGP link, it is propagated over internal links to other BGP-enabled devices within the AS.

BGP update messages include a MED metric if the route was learned from BGP and already had a MED metric associated with it, or if you configure the MED metric in the configuration file.

A MED metric is advertised with a route according to the following general rules:

- A more specific metric overrides a less specific metric. That is, a group-specific metric overrides a global BGP metric, and a peer-specific metric overrides a global BGP or group-specific metric.
- A metric defined with a routing policy overrides a metric defined with the **metric-out** statement.
- If any metric is defined, it overrides a metric received in a route.
- If the received route does not have an associated MED metric, and if you do not explicitly configure a metric value, no metric is advertised. When you do not explicitly configure a metric value, the MED value is equivalent to zero (0) when advertising an active route.

Because the AS path rather than the number of hops between hosts is the primary criterion for BGP route selection, an AS with multiple connections to a peer AS can have multiple equivalent AS paths. When the routing table contains two routes to the same host in a neighboring AS, an MED metric assigned to each route can determine which to include in the forwarding table. The MED metric you assign can force traffic through a particular exit point in an AS.

Figure 9 on page 71 illustrates how MED metrics are used to determine route selection.

Figure 9: Default MED Example

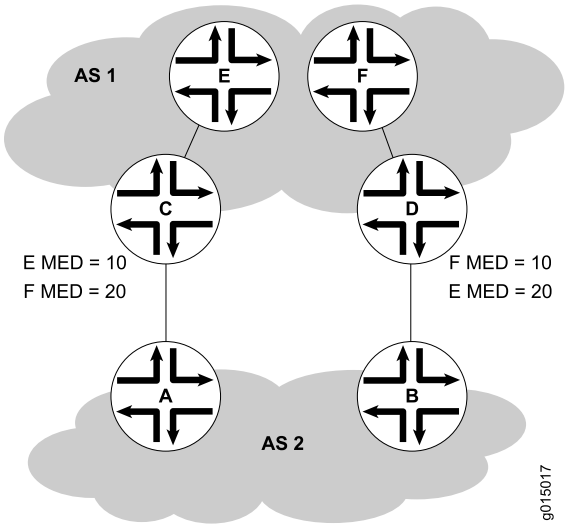


Figure 9 on page 71 shows AS 1 and AS 2 connected by two separate BGP links to Routers C and D. Host E in AS 1 is located nearer to Router C. Host F, also in AS 1, is located nearer to Router D. Because the AS paths are equivalent, two routes exist for each host, one through Router C and one through Router D. To force all traffic destined for Host E through Router C, the network administrator for AS 1 assigns an MED metric for each router to Host E at its exit point. An MED metric of 10 is assigned to the route to Host E through Router C, and an MED metric of 20 is assigned to the route to Host E through Router D. BGP routers in AS 2 then select the route with the lower MED metric for the forwarding table.

By default, only the MEDs of routes that have the same peer ASs are compared. However, you can configure the routing table path selection options listed in Table 3 on page 72 to compare MEDs in different ways. The MED options are not mutually exclusive and can be configured in combination or independently. For the MED options to take effect, you must configure them uniformly all through your network. The MED option or options you configure determine the route selected. Thus we recommend that you carefully evaluate your network for preferred routes before configuring the MED options.

Table 3: MED Options for Routing Table Path Selection

Option (Name)	Function	Use
Always comparing MEDs (**always-compare-med**)	Ensures that the MEDs for paths from peers in different ASs are always compared in the route selection process.	Useful when all enterprises participating in a network agree on a uniform policy for setting MEDs. For example, in a network shared by two ISPs, both must agree that a certain path is the better path to configure the MED values correctly.
Adding IGP cost to MED (**med-plus-igp**)	Before comparing MED values for path selection, adds to the MED the cost of the IGP route to the BGP next-hop destination. This option replaces the MED value for the router, but does not affect the IGP metric comparison. As a result, when multiple routes have the same value after the MED-plus-IPG comparison, and route selection continues, the IGP route metric is also compared, even though it was added to the MED value and compared earlier in the selection process.	Useful when the downstream AS requires the complete cost of a certain route that is received across multiple ASs.
Applying Cisco IOS nondeterministic behavior (**cisco-non-deterministic**)	Specifies the nondeterministic behavior of the Cisco IOS software: • The active path is always first. All nonactive but eligible paths follow the active path and are maintained in the order in which they were received. Ineligible paths remain at the end of the list. • When a new path is added to the routing table, path comparisons are made among all routes, including those paths that must never be selected because they lose the MED tie-breaking rule.	We recommend that you do not configure this option, because the nondeterministic behavior sometimes prevents the system from properly comparing the MEDs between paths.

Example: Configuring the MED Attribute Directly

This example shows how to configure a multiple exit discriminator (MED) metric to advertise in BGP update messages.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

To directly configure a MED metric to advertise in BGP update messages, include the **metric-out** statement:

```
metric-out (metric | minimum-igp offset | igp delay-med-update | offset);
```

metric is the primary metric on all routes sent to peers. It can be a value in the range from 0 through 4,294,967,295 ($2^{32} - 1$).

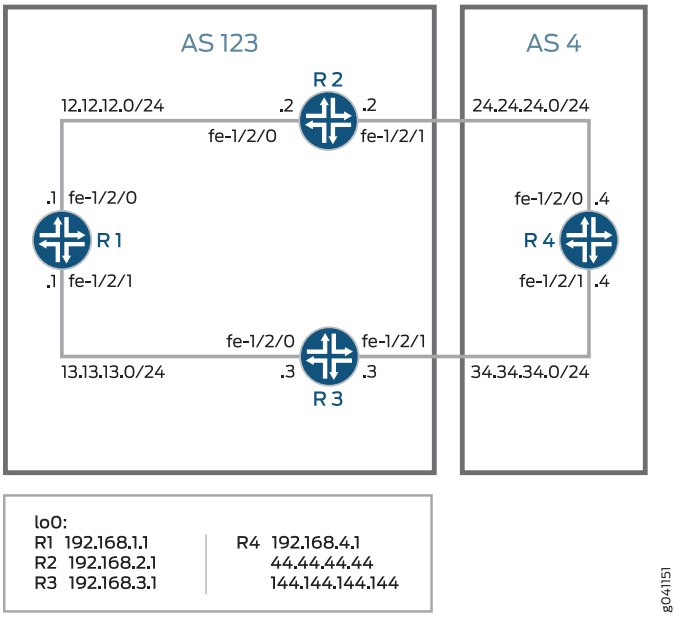
The following optional settings are also supported:

- **minimum-igp**—Sets the metric to the minimum metric value calculated in the interior gateway protocol (IGP) to get to the BGP next hop. If a newly calculated metric is greater than the minimum metric value, the metric value remains unchanged. If a newly calculated metric is lower, the metric value is lowered to that value.
- **igp**—Sets the metric to the most recent metric value calculated in the IGP to get to the BGP next hop.
- **delay-med-update**—Delays sending MED updates when the MED value increases. Include the **delay-med-update** statement when you configure the **igp** statement. The default interval to delay sending updates, unless the MED is lower or another attribute associated with the route has changed is 10 minutes. Include the **med-igp-update-interval *minutes*** statement at the **[edit routing-options]** hierarchy level to modify the default interval.
- ***offset***—Specifies a value for ***offset*** to increase or decrease the metric that is used from the metric value calculated in the IGP. The metric value is offset by the value specified. The metric calculated in the IGP (by specifying either **igp** or **igp-minimum**) is increased if the ***offset*** value is positive. The metric calculated in the IGP (by specifying either **igp** or **igp-minimum**) is decreased if the ***offset*** value is negative.

  offset can be a value in the range from -2^{31} through $2^{31} - 1$. Note that the adjusted metric can never go below 0 or above $2^{32} - 1$.

Figure 10 on page 74 shows a typical network with internal peer sessions and multiple exit points to a neighboring autonomous system (AS).

Figure 10: Typical Network with IBGP Sessions and Multiple Exit Points



Device R4 has multiple loopback interfaces configured to simulate advertised prefixes. The extra loopback interface addresses are 44.44.44.44/32 and 144.144.144.144/32. This example shows how to configure Device R4 to advertise a MED value of 30 to Device R3 and a MED value of 20 to Device R2. This causes all of the devices in AS 123 to prefer the path through Device R2 to reach AS 4.

Configuration

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 12.12.12.1/24
set interfaces fe-1/2/1 unit 2 family inet address 13.13.13.1/24
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.1.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
set protocols ospf area 0.0.0.0 interface fe-1/2/1.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
```

```
set routing-options autonomous-system 123
set routing-options router-id 192.168.1.1
```

Device R2

```
set interfaces fe-1/2/0 unit 3 family inet address 12.12.12.2/24
set interfaces fe-1/2/1 unit 4 family inet address 24.24.24.2/24
set interfaces lo0 unit 2 family inet address 192.168.2.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.2.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 24.24.24.4
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.2.1
```

Device R3

```
set interfaces fe-1/2/0 unit 5 family inet address 13.13.13.3/24
set interfaces fe-1/2/1 unit 6 family inet address 34.34.34.3/24
set interfaces lo0 unit 3 family inet address 192.168.3.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.3.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 34.34.34.4
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.5
set protocols ospf area 0.0.0.0 interface fe-1/2/1.6
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.3.1
```

Device R4

```
set interfaces fe-1/2/0 unit 7 family inet address 24.24.24.4/24
set interfaces fe-1/2/1 unit 8 family inet address 34.34.34.4/24
set interfaces lo0 unit 4 family inet address 192.168.4.1/32
set interfaces lo0 unit 4 family inet address 44.44.44.44/32
set interfaces lo0 unit 4 family inet address 144.144.144.144/32
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 123
set protocols bgp group external neighbor 34.34.34.3 metric-out 30
set protocols bgp group external neighbor 24.24.24.2 metric-out 20
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
```

```
set routing-options autonomous-system 4
set routing-options router-id 192.168.4.1
```

Configuring Device R1

Step-by-Step Procedure

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

To configure Device R1:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 1]
user@R1# set family inet address 12.12.12.1/24

[edit interfaces fe-1/2/1 unit 2]
user@R1# set family inet address 13.13.13.1/24

[edit interfaces lo0 unit 1]
user@R1# set family inet address 192.168.1.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R1# set type internal
user@R1# set local-address 192.168.1.1
user@R1# set export send-direct
user@R1# set neighbor 192.168.2.1
user@R1# set neighbor 192.168.3.1
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface lo0.1 passive
user@R1# set interface fe-1/2/0.1
user@R1# set interface fe-1/2/1.2
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R1# set from protocol direct
user@R1# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 123
user@R1# set router-id 192.168.1.1
```

Results

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

```
user@R1# show interfaces
fe-1/2/0 {
   unit 1 {
      family inet {
         address 12.12.12.1/24;
      }
   }
}
fe-1/2/1 {
   unit 2 {
      family inet {
         address 13.13.13.1/24;
      }
   }
}
lo0 {
   unit 1 {
      family inet {
         address 192.168.1.1/32;
      }
   }
}
```

```
user@R1# show policy-options
policy-statement send-direct {
   term 1 {
      from protocol direct;
      then accept;
   }
}
```

```
user@R1# show protocols
bgp {
   group internal {
      type internal;
      local-address 192.168.1.1;
      export send-direct;
      neighbor 192.168.2.1;
      neighbor 192.168.3.1;
   }
}
ospf {
   area 0.0.0.0 {
      interface lo0.1 {
         passive;
      }
      interface fe-1/2/0.1;
      interface fe-1/2/1.2;
   }
}
```

```
user@R1# show routing-options
autonomous-system 123;
router-id 192.168.1.1;
```

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

Configuring Device R2

Step-by-Step Procedure

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

To configure Device R2:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 3]
user@R2# set family inet address 12.12.12.21/24

[edit interfaces fe-1/2/1 unit 4]
user@R2# set family inet address 24.24.24.2/24

[edit interfaces lo0 unit 2]
user@R2# set family inet address 192.168.2.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R2# set type internal
user@R2# set local-address 192.168.2.1
user@R2# set export send-direct
user@R2# set neighbor 192.168.1.1
user@R2# set neighbor 192.168.3.1

[edit protocols bgp group external]
user@R2# set type external
user@R2# set export send-direct
user@R2# set peer-as 4
user@R2# set neighbor 24.24.24.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R2# set interface lo0.2 passive
user@R2# set interface fe-1/2/0.3
user@R2# set interface fe-1/2/1.4
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R2# set from protocol direct
user@R2# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R2# set autonomous-system 123
user@R2# set router-id 192.168.2.1
```

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

```
user@R2# show interfaces
fe-1/2/0 {
    unit 3 {
        family inet {
            address 12.12.12.2/24;
        }
    }
}
fe-1/2/1 {
    unit 4 {
        family inet {
            address 24.24.24.2/24;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 192.168.2.1/32;
        }
    }
}
```

```
user@R2# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}
```

```
user@R2# show protocols
bgp {
    group internal {
        type internal;
        local-address 192.168.2.1;
        export send-direct;
        neighbor 192.168.1.1;
        neighbor 192.168.3.1;
    }
    group external {
        type external;
        export send-direct;
        peer-as 4;
        neighbor 24.24.24.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.2 {
            passive;
```

```
    }
    interface fe-1/2/0.3;
    interface fe-1/2/1.4;
  }
}
```

```
user@R2# show routing-options
autonomous-system 123;
router-id 192.168.2.1;
```

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

Configuring Device R3

Step-by-Step Procedure

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

To configure Device R3:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 5]
user@R3# set family inet address 13.13.13.3/24

[edit interfaces fe-1/2/1 unit 6]
user@R3# set family inet address 34.34.34.3/24

[edit interfaces lo0 unit 3]
user@R3# set family inet address 192.168.3.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R3# set type internal
user@R3# set local-address 192.168.3.1
user@R3# set export send-direct
user@R3# set neighbor 192.168.1.1
user@R3# set neighbor 192.168.2.1

[edit protocols bgp group external]
user@R3# set type external
user@R3# set export send-direct
user@R3# set peer-as 4
user@R3# set neighbor 34.34.34.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R3# set interface lo0.3 passive
user@R3# set interface fe-1/2/0.5
user@R3# set interface fe-1/2/1.6
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R3# set from protocol direct
user@R3# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R3# set autonomous-system 123
user@R3# set router-id 192.168.3.1
```

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

```
user@R3# show interfaces
fe-1/2/0 {
  unit 5 {
    family inet {
      address 13.13.13.3/24;
    }
  }
}
fe-1/2/1 {
  unit 6 {
    family inet {
      address 34.34.34.3/24;
    }
  }
}
lo0 {
  unit 3 {
    family inet {
      address 192.168.3.1/32;
    }
  }
}
```

```
user@R3# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
```

```
user@R3# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.3.1;
    export send-direct;
    neighbor 192.168.1.1;
    neighbor 192.168.2.1;
  }
  group external {
```

```
      type external;
      export send-direct;
      peer-as 4;
      neighbor 34.34.34.4;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.3 {
        passive;
      }
      interface fe-1/2/0.5;
      interface fe-1/2/1.6;
    }
  }
```

```
user@R3# show routing-options
autonomous-system 123;
router-id 192.168.3.1;
```

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

Configuring Device R4

Step-by-Step Procedure

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

To configure Device R4:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 7]
user@R4# set family inet address 24.24.24.4/24

[edit interfaces fe-1/2/1 unit 8]
user@R4# set family inet address 34.34.34.4/24

[edit interfaces lo0 unit 4]
user@R4# set family inet address 192.168.4.1/32
user@R4# set family inet address 44.44.44.44/32
user@R4# set family inet address 144.144.144.144/32
```

   Device R4 has multiple loopback interface addresses to simulate advertised prefixes.

2. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R4# set from protocol direct
user@R4# set then accept
```

3. Configure BGP.

```
[edit protocols bgp group external]
user@R4# set type external
```

```
user@R4# set export send-direct
user@R4# set peer-as 123
```

4. Configure a MED value of 30 for neighbor Device R3, and a MED value of 20 for neighbor Device R2.

```
[edit protocols bgp group external]
user@R4# set neighbor 34.34.34.3 metric-out 30
user@R4# set neighbor 24.24.24.2 metric-out 20
```

This configuration causes autonomous system (AS) 123 (of which Device R1, Device R2, and Device R3 are members) to prefer the path through Device R2 to reach AS 4.

5. Configure the router ID and AS number.

```
[edit routing-options]
user@R4# set autonomous-system 4
user@R4# set router-id 192.168.4.1
```

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

```
user@R4# show interfaces
fe-1/2/0 {
    unit 7 {
        family inet {
            address 24.24.24.4/24;
        }
    }
}
fe-1/2/1 {
    unit 8 {
        family inet {
            address 34.34.34.4/24;
        }
    }
}
lo0 {
    unit 4 {
        family inet {
            address 192.168.4.1/32;
            address 44.44.44.44/32;
            address 144.144.144.144/32;
        }
    }
}
```

```
user@R4# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}
```

```
user@R4# show protocols
bgp {
  group external {
    type external;
    export send-direct;
    peer-as 123;
    neighbor 34.34.34.3 {
      metric-out 30;
    }
    neighbor 24.24.24.2 {
      metric-out 20;
    }
  }
}

user@R4# show routing-options
autonomous-system 4;
router-id 192.168.4.1;
```

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

Verification

Confirm that the configuration is working properly.

- Checking the Active Path From Device R1 to Device R4 on page 84
- Verifying That Device R4 Is Sending Its Routes Correctly on page 85

Checking the Active Path From Device R1 to Device R4

Purpose Verify that the active path goes through Device R2.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@R1> show route protocol bgp
inet.0: 13 destinations, 19 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

12.12.12.0/24       [BGP/170] 3d 22:52:38, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
13.13.13.0/24       [BGP/170] 3d 03:15:16, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
24.24.24.0/24       [BGP/170] 3d 22:52:38, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
34.34.34.0/24       [BGP/170] 3d 03:15:16, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
44.44.44.44/32     *[BGP/170] 01:41:11, MED 20, localpref 100, from 192.168.2.1
                      AS path: 4 I
                    > to 12.12.12.2 via fe-1/2/0.1
144.144.144.144/32 *[BGP/170] 00:08:13, MED 20, localpref 100, from 192.168.2.1
                      AS path: 4 I
                    > to 12.12.12.2 via fe-1/2/0.1
192.168.2.1/32      [BGP/170] 3d 22:52:38, localpref 100, from 192.168.2.1
                      AS path: I
```

```
                    > to 12.12.12.2 via fe-1/2/0.1
192.168.3.1/32      [BGP/170] 3d 03:15:16, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.4.1/32     *[BGP/170] 01:41:11, MED 20, localpref 100, from 192.168.2.1
                      AS path: 4 I
                    > to 12.12.12.2 via fe-1/2/0.1
```

Meaning The asterisk (*) shows that the preferred path is through Device R2. The reason for the path selection is listed as MED 20.

Verifying That Device R4 Is Sending Its Routes Correctly

Purpose Make sure that Device R4 is sending update messages with a value of 20 to Device R2 and a value of 30 to Device R3.

Action From operational mode, enter the **show route advertising-protocol bgp 24.24.24.2** command.

```
user@R4> show route advertising-protocol bgp 24.24.24.2
inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 24.24.24.0/24           Self                 20                 I
* 34.34.34.0/24           Self                 20                 I
* 44.44.44.44/32          Self                 20                 I
* 144.144.144.144/32      Self                 20                 I
* 192.168.4.1/32          Self                 20                 I

user@R4> show route advertising-protocol bgp 34.34.34.3
inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 24.24.24.0/24           Self                 30                 I
* 34.34.34.0/24           Self                 30                 I
* 44.44.44.44/32          Self                 30                 I
* 144.144.144.144/32      Self                 30                 I
* 192.168.4.1/32          Self                 30                 I
```

Meaning The MED column shows that Device R4 is sending the correct MED values to its two external BGP (EBGP) neighbors.

Example: Configuring the MED Using Route Filters

This example shows how to configure a policy that uses route filters to modify the multiple exit discriminator (MED) metric to advertise in BGP update messages.

Requirements

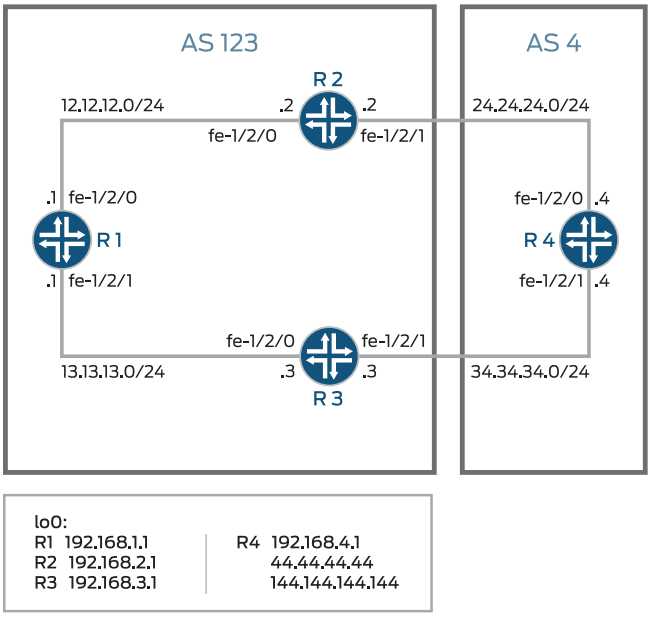
No special configuration beyond device initialization is required before you configure this example.

Overview

To configure a route-filter policy that modifies the advertised MED metric in BGP update messages, include the **metric** statement in the policy action.

Figure 11 on page 86 shows a typical network with internal peer sessions and multiple exit points to a neighboring autonomous system (AS).

Figure 11: Typical Network with IBGP Sessions and Multiple Exit Points



Device R4 has multiple loopback interfaces configured to simulate advertised prefixes. The extra loopback interface addresses are 44.44.44.44/32 and 144.144.144.144/32. This example shows how to configure Device R4 to advertise a MED value of 30 to Device R3 for all routes except 144.144.144.144. For 144.144.144.144, a MED value of 10 is advertised to Device 3. A MED value of 20 is advertised to Device R2, regardless of the route prefix.

Configuration

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 12.12.12.1/24
set interfaces fe-1/2/1 unit 2 family inet address 13.13.13.1/24
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.1.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
```

```
set protocols ospf area 0.0.0.0 interface fe-1/2/1.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.1.1
```

Device R2

```
set interfaces fe-1/2/0 unit 3 family inet address 12.12.12.2/24
set interfaces fe-1/2/1 unit 4 family inet address 24.24.24.2/24
set interfaces lo0 unit 2 family inet address 192.168.2.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.2.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.3.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 24.24.24.4
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.2.1
```

Device R3

```
set interfaces fe-1/2/0 unit 5 family inet address 13.13.13.3/24
set interfaces fe-1/2/1 unit 6 family inet address 34.34.34.3/24
set interfaces lo0 unit 3 family inet address 192.168.3.1/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.3.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.1.1
set protocols bgp group internal neighbor 192.168.2.1
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 4
set protocols bgp group external neighbor 34.34.34.4
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.5
set protocols ospf area 0.0.0.0 interface fe-1/2/1.6
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 123
set routing-options router-id 192.168.3.1
```

Device R4

```
set interfaces fe-1/2/0 unit 7 family inet address 24.24.24.4/24
set interfaces fe-1/2/1 unit 8 family inet address 34.34.34.4/24
set interfaces lo0 unit 4 family inet address 192.168.4.1/32
set interfaces lo0 unit 4 family inet address 44.44.44.44/32
set interfaces lo0 unit 4 family inet address 144.144.144.144/32
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 123
set protocols bgp group external neighbor 34.34.34.3 export med-10
```

```
set protocols bgp group external neighbor 34.34.34.3 export med-30
set protocols bgp group external neighbor 24.24.24.2 metric-out 20
set policy-options policy-statement med-10 from route-filter 144.144.144.144/32 exact
set policy-options policy-statement med-10 then metric 10
set policy-options policy-statement med-10 then accept
set policy-options policy-statement med-30 from route-filter 0.0.0.0/0 longer
set policy-options policy-statement med-30 then metric 30
set policy-options policy-statement med-30 then accept
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 4
set routing-options router-id 192.168.4.1
```

Configuring Device R1

Step-by-Step Procedure

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

To configure Device R1:

1. Configure the device interfaces.

```
[edit interfaces fe-1/2/0 unit 1]
user@R1# set family inet address 12.12.12.1/24

[edit interfaces fe-1/2/1 unit 2]
user@R1# set family inet address 13.13.13.1/24

[edit interfaces lo0 unit 1]
user@R1# set family inet address 192.168.1.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R1# set type internal
user@R1# set local-address 192.168.1.1
user@R1# set export send-direct
user@R1# set neighbor 192.168.2.1
user@R1# set neighbor 192.168.3.1
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface lo0.1 passive
user@R1# set interface fe-1/2/0.1
user@R1# set interface fe-1/2/1.2
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R1# set from protocol direct
user@R1# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 123
user@R1# set router-id 192.168.1.1
```

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 12.12.12.1/24;
    }
  }
}
fe-1/2/1 {
  unit 2 {
    family inet {
      address 13.13.13.1/24;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.1.1/32;
    }
  }
}
```

```
user@R1# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.1.1;
    export send-direct;
    neighbor 192.168.2.1;
    neighbor 192.168.3.1;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.1 {
      passive;
    }
    interface fe-1/2/0.1;
    interface fe-1/2/1.2;
  }
}
```

```
user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
```

```
        then accept;
    }
}
```

```
user@R1# show routing-options
autonomous-system 123;
router-id 192.168.1.1;
```

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

Configuring Device R2

Step-by-Step Procedure

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

To configure Device R2:

1. Configure the device interfaces.

```
[edit interfaces fe-1/2/0 unit 3]
user@R2# set family inet address 12.12.12.21/24

[edit interfaces fe-1/2/1 unit 4]
user@R2# set family inet address 24.24.24.2/24

[edit interfaces lo0 unit 2]
user@R2# set family inet address 192.168.2.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R2# set type internal
user@R2# set local-address 192.168.2.1
user@R2# set export send-direct
user@R2# set neighbor 192.168.1.1
user@R2# set neighbor 192.168.3.1

[edit protocols bgp group external]
user@R2# set type external
user@R2# set export send-direct
user@R2# set peer-as 4
user@R2# set neighbor 24.24.24.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R2# set interface lo0.2 passive
user@R2# set interface fe-1/2/0.3
user@R2# set interface fe-1/2/1.4
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R2# set from protocol direct
```

```
user@R2# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R2# set autonomous-system 123
user@R2# set router-id 192.168.2.1
```

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

```
user@R2# show interfaces
fe-1/2/0 {
    unit 3 {
        family inet {
            address 12.12.12.2/24;
        }
    }
}
fe-1/2/1 {
    unit 4 {
        family inet {
            address 24.24.24.2/24;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 192.168.2.1/32;
        }
    }
}
```

```
user@R2# show protocols
bgp {
    group internal {
        type internal;
        local-address 192.168.2.1;
        export send-direct;
        neighbor 192.168.1.1;
        neighbor 192.168.3.1;
    }
    group external {
        type external;
        export send-direct;
        peer-as 4;
        neighbor 24.24.24.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.2 {
            passive;
```

```
    }
    interface fe-1/2/0.3;
    interface fe-1/2/1.4;
  }
}
```

```
user@R2# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
```

```
user@R2# show routing-options
autonomous-system 123;
router-id 192.168.2.1;
```

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

Configuring Device R3

Step-by-Step Procedure

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

To configure Device R3:

1. Configure the device interfaces.

```
[edit interfaces fe-1/2/0 unit 5]
user@R3# set family inet address 13.13.13.3/24

[edit interfaces fe-1/2/1 unit 6]
user@R3# set family inet address 34.34.34.3/24

[edit interfaces lo0 unit 3]
user@R3# set family inet address 192.168.3.1/32
```

2. Configure BGP.

```
[edit protocols bgp group internal]
user@R3# set type internal
user@R3# set local-address 192.168.3.1
user@R3# set export send-direct
user@R3# set neighbor 192.168.1.1
user@R3# set neighbor 192.168.2.1

[edit protocols bgp group external]
user@R3# set type external
user@R3# set export send-direct
user@R3# set peer-as 4
user@R3# set neighbor 34.34.34.4
```

3. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
```

```
user@R3# set interface lo0.3 passive
user@R3# set interface fe-1/2/0.5
user@R3# set interface fe-1/2/1.6
```

4. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R3# set from protocol direct
user@R3# set then accept
```

5. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R3# set autonomous-system 123
user@R3# set router-id 192.168.3.1
```

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

```
user@R3# show interfaces
fe-1/2/0 {
  unit 5 {
    family inet {
      address 13.13.13.3/24;
    }
  }
}
fe-1/2/1 {
  unit 6 {
    family inet {
      address 34.34.34.3/24;
    }
  }
}
lo0 {
  unit 3 {
    family inet {
      address 192.168.3.1/32;
    }
  }
}
```

```
user@R3# show protocols
bgp {
  group internal {
    type internal;
    local-address 192.168.3.1;
    export send-direct;
    neighbor 192.168.1.1;
    neighbor 192.168.2.1;
  }
  group external {
```

```
      type external;
      export send-direct;
      peer-as 4;
      neighbor 34.34.34.4;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.3 {
        passive;
      }
      interface fe-1/2/0.5;
      interface fe-1/2/1.6;
    }
  }
```

```
user@R3# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
```

```
user@R3# show routing-options
autonomous-system 123;
router-id 192.168.3.1;
```

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

Configuring Device R4

Step-by-Step Procedure

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

To configure Device R4:

1. Configure the device interfaces.

```
[edit interfaces fe-1/2/0 unit 7]
user@R4# set family inet address 24.24.24.4/24
```

```
[edit interfaces fe-1/2/1 unit 8]
user@R4# set family inet address 34.34.34.4/24
```

```
[edit interfaces lo0 unit 4]
user@R4# set family inet address 192.168.4.1/32
user@R4# set family inet address 44.44.44.44/32
user@R4# set family inet address 144.144.144.144/32
```

   Device R4 has multiple loopback interface addresses to simulate advertised prefixes.

2. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 1]
user@R4# set from protocol direct
user@R4# set then accept
```

3. Configure BGP.

```
[edit protocols bgp group external]
user@R4# set type external
user@R4# set export send-direct
user@R4# set peer-as 123
```

4. Configure the two MED policies.

```
[edit policy-options]
set policy-statement med-10 from route-filter 144.144.144.144/32 exact
set policy-statement med-10 then metric 10
set policy-statement med-10 then accept


set policy-statement med-30 from route-filter 0.0.0.0/0 longer
set policy-statement med-30 then metric 30
set policy-statement med-30 then accept
```

5. Configure the two EBGP neighbors, applying the two MED policies to Device R3, and a MED value of 20 to Device R2.

```
[edit protocols bgp group external]
user@R4# set neighbor 34.34.34.3 export med-10
user@R4# set neighbor 34.34.34.3 export med-30
user@R4# set neighbor 24.24.24.2 metric-out 20
```

6. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@R4# set autonomous-system 4
user@R4# set router-id 192.168.4.1
```

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

```
user@R4# show interfaces
fe-1/2/0 {
  unit 7 {
    family inet {
      address 24.24.24.4/24;
    }
  }
}
fe-1/2/1 {
  unit 8 {
    family inet {
      address 34.34.34.4/24;
    }
  }
}
lo0 {
```

```
    unit 4 {
      family inet {
        address 192.168.4.1/32;
        address 44.44.44.44/32;
        address 144.144.144.144/32;
      }
    }
}
```

```
user@R4# show protocols
bgp {
  group external {
    type external;
    export send-direct;
    peer-as 123;
    neighbor 24.24.24.2 {
      metric-out 20;
    }
    neighbor 34.34.34.3 {
      export [ med-10 med-30 ];
    }
  }
}
```

```
user@R4# show policy-options
policy-statement med-10 {
  from {
    route-filter 144.144.144.144/32 exact;
  }
  then {
    metric 10;
    accept;
  }
}
policy-statement med-30 {
  from {
    route-filter 0.0.0.0/0 longer;
  }
  then {
    metric 30;
    accept;
  }
}
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
```

```
user@R4# show routing-options
autonomous-system 4;
router-id 192.168.4.1;
```

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

Verification

Confirm that the configuration is working properly.

Checking the Active Path from Device R1 to Device R4

Purpose Verify that the active path goes through Device R2.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@R1> show route protocol bgp
inet.0: 13 destinations, 19 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

12.12.12.0/24       [BGP/170] 4d 01:13:32, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
13.13.13.0/24       [BGP/170] 3d 05:36:10, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
24.24.24.0/24       [BGP/170] 4d 01:13:32, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
34.34.34.0/24       [BGP/170] 3d 05:36:10, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
44.44.44.44/32     *[BGP/170] 00:06:03, MED 20, localpref 100, from 192.168.2.1
                      AS path: 4 I
                    > to 12.12.12.2 via fe-1/2/0.1
144.144.144.144/32 *[BGP/170] 00:06:03, MED 10, localpref 100, from 192.168.3.1
                      AS path: 4 I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.2.1/32      [BGP/170] 4d 01:13:32, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 12.12.12.2 via fe-1/2/0.1
192.168.3.1/32      [BGP/170] 3d 05:36:10, localpref 100, from 192.168.3.1
                      AS path: I
                    > to 13.13.13.3 via fe-1/2/1.2
192.168.4.1/32     *[BGP/170] 00:06:03, MED 20, localpref 100, from 192.168.2.1
                      AS path: 4 I
                    > to 12.12.12.2 via fe-1/2/0.1
```

Meaning The output shows that the preferred path to the routes advertised by Device R4 is through Device R2 for all routes except 144.144.144.144/32. For 144.144.144.144/32, the preferred path is through Device R3.

Verifying That Device R4 Is Sending Its Routes Correctly

Purpose Make sure that Device R4 is sending update messages with a value of 20 to Device R2 and a value of 30 to Device R3.

Action From operational mode, enter the **show route advertising-protocol bgp** command.

```
user@R4> show route advertising-protocol bgp 24.24.24.2
```

```
inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 24.24.24.0/24           Self                 20                 I
* 34.34.34.0/24           Self                 20                 I
* 44.44.44.44/32          Self                 20                 I
* 144.144.144.144/32      Self                 20                 I
* 192.168.4.1/32          Self                 20                 I

user@R4> show route advertising-protocol bgp 34.34.34.3
inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 24.24.24.0/24           Self                 30                 I
* 34.34.34.0/24           Self                 30                 I
* 44.44.44.44/32          Self                 30                 I
* 144.144.144.144/32      Self                 10                 I
* 192.168.4.1/32          Self                 30                 I
```

Meaning The MED column shows that Device R4 is sending the correct MED values to its two EBGP neighbors.

Example: Configuring the MED Using Communities

Set the multiple exit discriminator (MED) metric to 20 for all routes from a particular community.

```
[edit]
routing-options {
  router-id 10.0.0.1;
  autonomous-system 23;
}
policy-options {
  policy-statement from-otago {
    from community otago;
    then metric 20;
  }
  community otago members [56:2379 23:46944];
}
protocols {
  bgp {
    import from-otago;
    group 23 {
      type external;
      peer-as 56;
      neighbor 192.168.0.1 {
        traceoptions {
          file bgp-log-peer;
          flag packets;
        }
        log-updown;
      }
    }
  }
}
```

Example: Associating the MED Path Attribute with the IGP Metric and Delaying MED Updates

This example shows how to associate the multiple exit discriminator (MED) path attribute with the interior gateway protocol (IGP) metric, and configure a timer to delay update of the MED attribute.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

BGP can be configured to advertise the MED attribute for a route based on the IGP distance of its internal BGP (IBGP) route next-hop. The IGP metric enables internal routing to follow the shortest path according to the administrative setup. In some deployments, it might be ideal to communicate IGP shortest-path knowledge to external BGP (EBGP) peers in a neighboring autonomous system (AS). This allows those EBGP peers to forward traffic into your AS using the shortest paths possible.

Routes learned from an EBGP peer usually have a next hop on a directly connected interface, and thus the IGP value is equal to zero. Zero is the value advertised. The IGP metric is a nonzero value when a BGP peer sends third-party next hops that require the local system to perform next-hop resolution—IBGP configurations, configurations within confederation peers, or EBGP configurations that include the **multihop** command. In these scenarios, it might make sense to associate the MED value with the IGP metric by including the metric-out **minimum-igp** or metric-out **igp** option.

The drawback of associating the MED with the IGP metric is the risk of excessive route advertisements when there are IGP instabilities in the network. Configuring a delay for the MED update provides a mechanism to reduce route advertisements in such scenarios. The delay works by slowing down MED updates when the IGP metric for the next hop changes. The approach uses a timer to periodically advertise MED updates. When the timer expires, the MED attribute for routes with **metric-out igp delay-updates** configured is updated to the current IGP metric of the next hop. The BGP-enabled device sends out advertisements for routes for which the MED attribute has changed.

The **delay-updates** option identifies the BGP groups (or peers) for which the MED updates must be suppressed. The time for advertising MED updates is set to 10 minutes by default. You can increase the interval up to 600 minutes by including the **med-igp-update-interval** statement in the **routing-options** configuration.

NOTE: If you have nonstop active routing (NSR) enabled and a switchover occurs, the delayed MED updates might be advertised as soon as the switchover occurs.

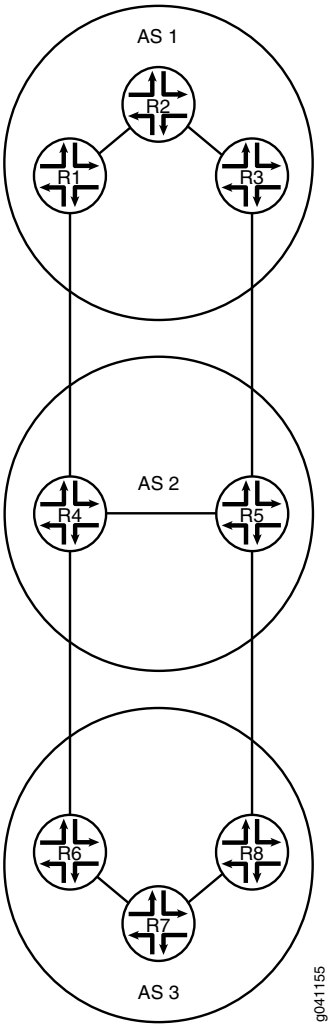
When you configure the **metric-out igp** option, the IGP metric directly tracks the IGP cost to the IBGP peer. When the IGP cost goes down, so does the advertised MED value. Conversely, when the IGP cost goes up, the MED value goes up as well.

When you configure the **metric-out minimum-igp** option, the advertised MED value changes only when the IGP cost to the IBGP peer goes down. An increase in the IGP cost does not affect the MED value. The router monitors and remembers the lowest IGP cost until the routing process (rpd) is restarted. The BGP peer sends an update only if the MED is lower than the previously advertised value or another attribute associated with the route has changed, or if the BGP peer is responding to a refresh route request.

This example uses the **metric** statement in the OSPF configuration to demonstrate that when the IGP metric changes, the MED also changes after the configured delay interval. The OSPF metric can range from 1 through 65,535.

Figure 12 on page 101 shows the sample topology.

Figure 12: Topology for Delaying the MED Update



In this example, the MED value advertised by Device R1 is associated with the IGP running in AS 1. The MED value advertised by Device R1 impacts the decisions of the neighboring AS (AS 2) when AS 2 is forwarding traffic into AS 1.

Configuration

- Configuring Device R1 on page 105

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 2 description R1->R2
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.1/30
set interfaces fe-1/2/1 unit 7 description R1->R4
set interfaces fe-1/2/1 unit 7 family inet address 172.16.0.1/30
set interfaces lo0 unit 1 family inet address 192.168.0.1/32
```

```
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.1
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.2
set protocols bgp group internal neighbor 192.168.0.3
set protocols bgp group external type external
set protocols bgp group external metric-out igp delay-med-update
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 2
set protocols bgp group external neighbor 172.16.0.2
set protocols ospf area 0.0.0.0 interface fe-1/2/0.2 metric 600
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options med-igp-update-interval 12
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 1
```

Device R2

```
set interfaces fe-1/2/0 unit 1 description R2->R1
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 4 description R2->R3
set interfaces fe-1/2/1 unit 4 family inet address 10.0.2.2/30
set interfaces lo0 unit 2 family inet address 192.168.0.2/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.2
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.1
set protocols bgp group internal neighbor 192.168.0.3
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.2
set routing-options autonomous-system 1
```

Device R3

```
set interfaces fe-1/2/0 unit 3 description R3->R2
set interfaces fe-1/2/0 unit 3 family inet address 10.0.2.1/30
set interfaces fe-1/2/1 unit 5 description R3->R5
set interfaces fe-1/2/1 unit 5 family inet address 172.16.0.5/30
set interfaces lo0 unit 3 family inet address 192.168.0.3/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.3
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.1
set protocols bgp group internal neighbor 192.168.0.2
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 2
set protocols bgp group external neighbor 172.16.0.6
set protocols ospf area 0.0.0.0 interface fe-1/2/0.3
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.3
```

```
set routing-options autonomous-system 1
```

Device R4

```
set interfaces fe-1/2/0 unit 8 description R4->R1
set interfaces fe-1/2/0 unit 8 family inet address 172.16.0.2/30
set interfaces fe-1/2/1 unit 9 description R4->R5
set interfaces fe-1/2/1 unit 9 family inet address 10.0.4.1/30
set interfaces fe-1/2/2 unit 13 description R4->R6
set interfaces fe-1/2/2 unit 13 family inet address 172.16.0.9/30
set interfaces lo0 unit 4 family inet address 192.168.0.4/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.4
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.5
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external neighbor 172.16.0.10 peer-as 3
set protocols bgp group external neighbor 172.16.0.1 peer-as 1
set protocols ospf area 0.0.0.0 interface fe-1/2/1.9
set protocols ospf area 0.0.0.0 interface lo0.4 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.4
set routing-options autonomous-system 2
```

Device R5

```
set interfaces fe-1/2/0 unit 6 description R5->R3
set interfaces fe-1/2/0 unit 6 family inet address 172.16.0.6/30
set interfaces fe-1/2/1 unit 10 description R5->R4
set interfaces fe-1/2/1 unit 10 family inet address 10.0.4.2/30
set interfaces fe-1/2/2 unit 11 description R5->R8
set interfaces fe-1/2/2 unit 11 family inet address 172.16.0.13/30
set interfaces lo0 unit 5 family inet address 192.168.0.5/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.5
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.4
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external neighbor 172.16.0.5 peer-as 1
set protocols bgp group external neighbor 172.16.0.14 peer-as 3
set protocols ospf area 0.0.0.0 interface fe-1/2/1.10
set protocols ospf area 0.0.0.0 interface lo0.5 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.5
set routing-options autonomous-system 2
```

Device R6

```
set interfaces fe-1/2/0 unit 14 description R6->R4
set interfaces fe-1/2/0 unit 14 family inet address 172.16.0.10/30
set interfaces fe-1/2/1 unit 15 description R6->R7
set interfaces fe-1/2/1 unit 15 family inet address 10.0.6.1/30
set interfaces lo0 unit 6 family inet address 192.168.0.6/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.6
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.7
```

```
set protocols bgp group internal neighbor 192.168.0.8
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 2
set protocols bgp group external neighbor 172.16.0.9 peer-as 2
set protocols ospf area 0.0.0.0 interface fe-1/2/1.15
set protocols ospf area 0.0.0.0 interface lo0.6 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.6
set routing-options autonomous-system 3
```

Device R7

```
set interfaces fe-1/2/0 unit 16 description R7->R6
set interfaces fe-1/2/0 unit 16 family inet address 10.0.6.2/30
set interfaces fe-1/2/1 unit 17 description R7->R8
set interfaces fe-1/2/1 unit 17 family inet address 10.0.7.2/30
set interfaces lo0 unit 7 family inet address 192.168.0.7/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.7
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.6
set protocols bgp group internal neighbor 192.168.0.8
set protocols ospf area 0.0.0.0 interface fe-1/2/0.16
set protocols ospf area 0.0.0.0 interface fe-1/2/1.17
set protocols ospf area 0.0.0.0 interface lo0.7 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.7
set routing-options autonomous-system 3
```

Device R8

```
set interfaces fe-1/2/0 unit 12 description R8->R5
set interfaces fe-1/2/0 unit 12 family inet address 172.16.0.14/30
set interfaces fe-1/2/1 unit 18 description R8->R7
set interfaces fe-1/2/1 unit 18 family inet address 10.0.7.1/30
set interfaces lo0 unit 8 family inet address 192.168.0.8/32
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.0.8
set protocols bgp group internal export send-direct
set protocols bgp group internal neighbor 192.168.0.6
set protocols bgp group internal neighbor 192.168.0.7
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 2
set protocols bgp group external neighbor 172.16.0.13 peer-as 2
set protocols ospf area 0.0.0.0 interface fe-1/2/1.18
set protocols ospf area 0.0.0.0 interface lo0.8 passive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options router-id 192.168.0.8
set routing-options autonomous-system 3
```

Configuring Device R1

Step-by-Step Procedure

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

To configure Device R1:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 2]
user@R1# set description R1->R2
user@R1# set family inet address 10.0.0.1/30

[edit interfaces fe-1/2/1 unit 7]
user@R1# set description R1->R4
user@R1# set family inet address 172.16.0.1/30

[edit interfaces lo0 unit 1]
user@R1# set family inet address 192.168.0.1/32
```

2. Configure IBGP.

```
[edit protocols bgp group internal]
user@R1# set type internal
user@R1# set local-address 192.168.0.1
user@R1# set export send-direct
user@R1# set neighbor 192.168.0.2
user@R1# set neighbor 192.168.0.3
```

3. Configure EBGP.

```
[edit protocols bgp group external]
user@R1# set type external
user@R1# set export send-direct
user@R1# set peer-as 2
user@R1# set neighbor 172.16.0.2
```

4. Associate the MED value with the IGP metric.

```
[edit protocols bgp group external]
user@R1# set metric-out igp delay-med-update
```

The default for the MED update is 10 minutes when you include the **delay-med-update** option. When you exclude the **delay-med-update** option, the MED update occurs immediately after the IGP metric changes.

5. (Optional) Configure the update interval for the MED update.

```
[edit routing-options]
user@R1# set med-igp-update-interval 12
```

You can configure the interval from 10 minutes through 600 minutes.

6. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface fe-1/2/0.2 metric 600
user@R1# set interface lo0.1 passive
```

The **metric** statement is used here to demonstrate what happens when the IGP metric changes.

7. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

   ```
   [edit policy-options policy-statement send-direct term 1]
   user@R1# set from protocol direct
   user@R1# set then accept
   ```

8. Configure the router ID and autonomous system (AS) number.

   ```
   [edit routing-options]
   user@R1# set router-id 192.168.0.1
   user@R1# set autonomous-system 1
   ```

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

```
user@R1# show interfaces
fe-1/2/0 {
    unit 2 {
        description R1->R2;
        family inet {
            address 10.0.0.1/30;
        }
    }
}
fe-1/2/1 {
    unit 7 {
        description R1->R4;
        family inet {
            address 172.16.0.1/30;
        }
    }
}
lo0 {
    unit 1 {
        family inet {
            address 192.168.0.1/32;
        }
    }
}

user@R1# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}

user@R1# show protocols
```

```
bgp {
  group internal {
    type internal;
    local-address 192.168.0.1;
    export send-direct;
    neighbor 192.168.0.2;
    neighbor 192.168.0.3;
  }
  group external {
    type external;
    metric-out igp delay-med-update;
    export send-direct;
    peer-as 2;
    neighbor 172.16.0.2;
  }
}
ospf {
  area 0.0.0.0 {
    interface fe-1/2/0.2 {
      metric 600;
    }
    interface lo0.1 {
      passive;
    }
  }
}
```

```
user@R1# show routing-options
med-igp-update-interval 12;
router-id 192.168.0.1;
autonomous-system 1;
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat the configuration steps on the other devices in the topology, as needed for your network.

Verification

Confirm that the configuration is working properly.

- Checking the BGP Advertisements on page 107
- Verifying That the MED Value Changes When the OSPF Metric Changes on page 108
- Testing the minimum-igp Setting on page 108

Checking the BGP Advertisements

Purpose Verify that Device R1 is advertising to Device R4 a BGP MED value that reflects the IGP metric.

Action From operational mode, enter the **show route advertising-protocol bgp** command.

```
user@R1> show route advertising-protocol bgp 172.16.0.2
inet.0: 19 destinations, 33 routes (19 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.0/30             Self                 0                  I
* 172.16.0.0/30           Self                 0                  I
```

```
* 172.16.0.4/30          Self                601                I
* 192.168.0.1/32         Self                0                  I
```

Meaning The 601 value in the MED column shows that the MED value has been updated to reflect the configured OSPF metric.

Verifying That the MED Value Changes When the OSPF Metric Changes

Purpose Make sure that when you raise the OSPF metric to 700, the MED value is updated to reflect this change.

Action From configuration mode, enter the **set protocols ospf area 0 interface fe-1/2/0.2 metric 700** command.

```
user@R1# set protocols ospf area 0 interface fe-1/2/0.2 metric 700
user@R1# commit
```

After waiting 12 minutes (the configured delay period), enter the **show route advertising-protocol bgp** command from operational mode.

```
user@R1> show route advertising-protocol bgp 172.16.0.2
inet.0: 19 destinations, 33 routes (19 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.0/30             Self                 0                  I
* 172.16.0.0/30           Self                 0                  I
* 172.16.0.4/30           Self                 701                I
* 192.168.0.1/32          Self                 0                  I
```

Meaning The 701 value in the MED column shows that the MED value has been updated to reflect the configured OSPF metric.

Testing the minimum-igp Setting

Purpose Change the configuration to use the **minimum-igp** statement instead of the **igp** statement. When you increase the OSPF metric, the MED value remains unchanged, but when you decrease the OSPF metric, the MED value reflects the new OSPF metric.

Action From configuration mode, delete the **igp** statement, add the **minimum-igp** statement, and increase the OSPF metric.

```
user@R1# delete protocols bgp group external metric-out igp
user@R1# set protocols bgp group external metric-out minimum-igp
user@R1# set protocols ospf area 0 interface fe-1/2/0.2 metric 800
user@R1# commit
```

From operational mode, enter the **show route advertising-protocol bgp** command to make sure that the MED value does not change.

```
user@R1> show route advertising-protocol bgp 172.16.0.2
inet.0: 19 destinations, 33 routes (19 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.0/30             Self                 0                  I
* 172.16.0.0/30           Self                 0                  I
* 172.16.0.4/30           Self                 701                I
* 192.168.0.1/32          Self                 0                  I
```

From configuration mode, decrease the OSPF metric.

```
user@R1# set protocols ospf area 0 interface fe-1/2/0.2 metric 20
user@R1# commit
```

From operational mode, enter the **show route advertising-protocol bgp** command to make sure that the MED value does change.

```
user@R1> show route advertising-protocol bgp 172.16.0.2
inet.0: 19 destinations, 33 routes (19 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.0/30             Self                 0                  I
* 172.16.0.0/30           Self                 0                  I
* 172.16.0.4/30           Self                 21                 I
* 192.168.0.1/32          Self                 0                  I
```

Meaning When the **minimum-igp** statement is configured, the MED value changes only when a shorter path is available.

Related Documentation

- Examples: Configuring External BGP Peering on page 11
- *BGP Configuration Overview*

Examples: Configuring BGP Local AS

Understanding the BGP Local AS Attribute

When an Internet service provider (ISP) acquires a network that belongs to a different autonomous system (AS), there is no seamless method for moving the BGP peers of the acquired network to the AS of the acquiring ISP. The process of configuring the BGP peers with the new AS number can be time-consuming and cumbersome. Sometimes customers do not want to or are not immediately able to modify their peer arrangements or configuration. During this kind of transition period, it can be useful to configure BGP-enabled devices in the new AS to use the former AS number in BGP updates. This former AS number is called a *local AS*.

Using a local AS number permits the routing devices in an acquired network to appear to belong to the former AS.

For example, ISP A, with an AS of 200, acquires ISP B, with an AS of 250. ISP B has a customer, ISP C, that does not want to change its configuration. After ISP B becomes part of ISP A, a local AS number of 250 is configured for use in EBGP peer sessions with ISP C. Consequently, the local AS number of 250 is either prepended before or used instead of the global AS number of 200 in the AS path used to export routes to direct external peers in ISP C.

If the route is received from an internal BGP (IBGP) peer, the AS path includes the local AS number prepended before the global AS number.

The local AS number is used instead of the global AS number if the route is an external route, such as a static route or an interior gateway protocol (IGP) route that is imported into BGP. If the route is external and you want the global AS number to be included in the AS path, you can apply a routing policy that uses **as-path-expand** or **as-path-prepend**. Use the **as-path-expand** policy action to place the global AS number behind the local AS number. Use the **as-path-prepend** policy action to place the global AS number in front of the local AS number.

For example:

```
user@R2# show policy-options
policy-statement prepend-global {
    term 1 {
        from protocol static;
        then {
            as-path-prepend 200; # or use as-path-expand
            accept;
        }
    }
}

user@R2# show protocols bgp
group ext {
    export prepend-global;
    type external;
    local-as 250;
    neighbor 10.0.0.1 {
        peer-as 100;
    }
    neighbor 10.1.0.2 {
        peer-as 300;
    }
}

user@R2# show routing-options
static {
    route 1.1.1.1/32 next-hop 10.0.0.1;
}
autonomous-system 200;
```

```
user@R3#  run show route 1.1.1.1 protocol bgp
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1/32          *[BGP/170] 00:05:11, localpref 100
                       AS path: 200 250 I, validation-state: unverified
                     > to 10.1.0.1 via lt-1/2/0.4
```

In a Layer 3 VPN scenario, in which a provider edge (PE) device uses external BGP (EBGP) to peer with a customer edge (CE) device, the **local-as** statement behaves differently than in the non-VPN scenario. In the VPN scenario, the global AS number defined in the master instance is prepended to the AS path by default. To override this behavior, you can configure the **no-prepend-global-as** in the routing-instance BGP configuration on the PE device, as shown here:

```
user@R2# show routing-instances
```

```
red {
    instance-type vrf;
    interface fe-1/2/0.2;
    route-distinguisher 2:1;
    vrf-target target:2:1;
    protocols {
        bgp {
            group toR1 {
                type external;
                peer-as 1;
                local-as 200 no-prepend-global-as;
                neighbor 10.1.1.1;
            }
        }
    }
}
```

The Junos operating system (Junos OS) implementation of the local AS attribute supports the following options:

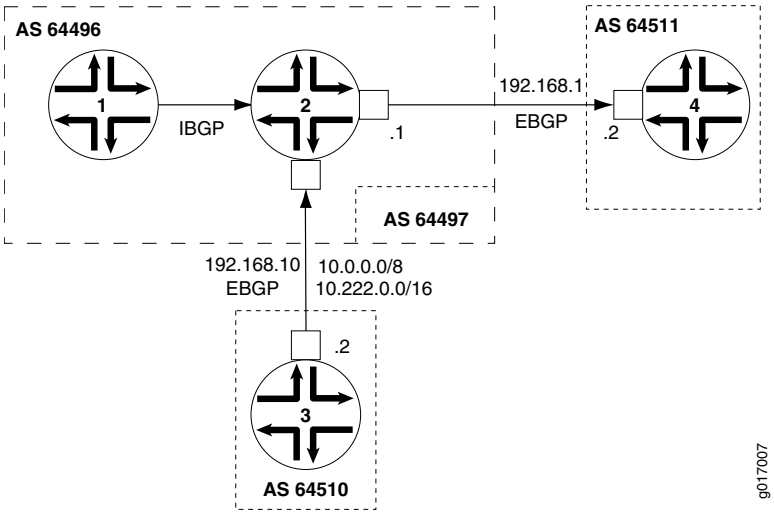
- **Local AS with private option**—When you use the **private** option, the local AS is used during the establishment of the BGP session with an EBGP neighbor but is hidden in the AS path sent to other EBGP peers. Only the global AS is included in the AS path sent to external peers.

  The **private** option is useful for establishing local peering with routing devices that remain configured with their former AS or with a specific customer that has not yet modified its peer arrangements. The local AS is used to establish the BGP session with the EBGP neighbor but is hidden in the AS path sent to external peers in another AS.

  Include the **private** option so that the local AS is not prepended before the global AS in the AS path sent to external peers. When you specify the **private** option, the local AS is prepended only in the AS path sent to the EBGP neighbor.

  For example, in Figure 13 on page 112, Router 1 and Router 2 are in AS 64496, Router 4 is in AS 64511, and Router 3 is in AS 64510. Router 2 formerly belonged to AS 64497, which has merged with another network and now belongs to AS 64496. Because Router 3 still peers with Router 2 using its former AS (64497), Router 2 needs to be configured with a local AS of 64497 in order to maintain peering with Router 3. Configuring a local AS of 64497 permits Router 2 to add AS 64497 when advertising routes to Router 3. Router 3 sees an AS path of 64497 64496 for the prefix 10/8.

Figure 13: Local AS Configuration



  To prevent Router 2 from adding the local AS number in its announcements to other peers, use the **local-as 64497 private** statement. This statement configures Router 2 to not include local AS 64497 when announcing routes to Router 1 and to Router 4. In this case, Router 4 sees an AS path of 64496 64510 for the prefix 10.222/16.

- **Local AS with alias option**—In Junos OS Release 9.5 and later, you can configure a local AS as an alias. During the establishment of the BGP open session, the AS used in the open message alternates between the local AS and the global AS. If the local AS is used to connect with the EBGP neighbor, then only the local AS is prepended to the AS path when the BGP peer session is established. If the global AS is used to connect with the EBGP neighbor, then only the global AS is prepended to the AS path when the BGP peer session is established. The use of the **alias** option also means that

the local AS is not prepended to the AS path for any routes learned from that EBGP neighbor. Therefore, the local AS remains hidden from other external peers.

Configuring a local AS with the **alias** option is especially useful when you are migrating the routing devices in an acquired network to the new AS. During the migration process, some routing devices might be configured with the new AS while others remain configured with the former AS. For example, it is good practice to start by first migrating to the new AS any routing devices that function as route reflectors. However, as you migrate the route reflector clients incrementally, each route reflector has to peer with routing devices configured with the former AS, as well as peer with routing devices configured with the new AS. To establish local peer sessions, it can be useful for the BGP peers in the network to use both the local AS and the global AS. At the same time, you want to hide this local AS from external peers and use only the global AS in the AS path when exporting routes to another AS. In this kind of situation, configure the **alias** option.

Include the **alias** option to configure the local AS as an alias to the global AS configured at the **[edit routing-options]** hierarchy level. When you configure a local AS as an alias, during the establishment of the BGP open session, the AS used in the open message alternates between the local AS and the global AS. The local AS is prepended to the AS path only when the peer session with an EBGP neighbor is established using that local AS. The local AS is hidden in the AS path sent to any other external peers. Only the global AS is prepended to the AS path when the BGP session is established using the global AS.

NOTE: The private and alias options are mutually exclusive. You cannot configure both options with the same local-as statement.

- **Local AS with option not to prepend the global AS**—In Junos OS Release 9.6 and later, you can configure a local AS with the option not to prepend the global AS. Only the local AS is included in the AS path sent to external peers.

  Use the **no-prepend-global-as** option when you want to strip the global AS number from outbound BGP updates in a virtual private network (VPN) scenario. This option is useful in aVPN scenario in which you want to hide the global AS from the VPN.

  Include the **no-prepend-global-as** option to have the global AS configured at the **[edit routing-options]** hierarchy level removed from the AS path sent to external peers. When you use this option, only the local AS is included in the AS path for the routes sent to a customer edge (CE) device.

- **Number of loops option**—The local AS feature also supports specifying the number of times that detection of the AS number in the AS_PATH attribute causes the route to be discarded or hidden. For example, if you configure **loops 1**, the route is hidden if the AS number is detected in the path one or more times. This is the default behavior. If you configure **loops 2**, the route is hidden if the AS number is detected in the path two or more times.

  For the **loops *number*** statement, you can configure 1 through 10.

NOTE: If you configure the local AS values for any BGP group, the detection of routing loops is performed using both the AS and the local AS values for all BGP groups.

If the local AS for the EBGP or IBGP peer is the same as the current AS, do not use the local-as statement to specify the local AS number.

When you configure the local AS within a VRF, this impacts the AS path loop-detection mechanism. All of the local-as statements configured on the device are part of a single AS domain. The AS path loop-detection mechanism is based on looking for a matching AS present in the domain.

Example: Configuring a Local AS for EBGP Sessions

This example shows how to configure a local autonomous system (AS) for a BGP peer so that both the global AS and the local AS are used in BGP inbound and outbound updates.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

Use the **local-as** statement when ISPs merge and want to preserve a customer's configuration, particularly the AS with which the customer is configured to establish a peer relationship. The **local-as** statement simulates the AS number already in place in customer routers, even if the ISP's router has moved to a different AS.

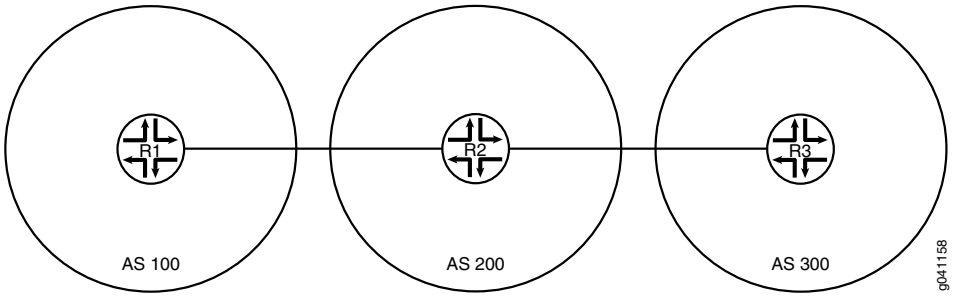
This example shows how to use the **local-as** statement to configure a local AS. The **local-as** statement is supported for BGP at the global, group, and neighbor hierarchy levels.

When you configure the **local-as** statement, you must specify an AS number. You can specify a number from 1 through 4,294,967,295 in plain-number format. In Junos OS Release 9.1 and later, the range for AS numbers is extended to provide BGP support for 4-byte AS numbers as defined in RFC 4893, *BGP Support for Four-octet AS Number Space*. In Junos OS Release 9.3 and later, you can also configure a 4-byte AS number using the AS-dot notation format of two integer values joined by a period: *<16-bit high-order value in decimal>.<16-bit low-order value in decimal>*. For example, the 4-byte AS number of 65,546 in plain-number format is represented as 1.10 in the AS-dot notation format. You can specify a value from 0.0 through 65535.65535 in AS-dot notation format. Junos

OS continues to support 2-byte AS numbers. The 2-byte AS number range is 1 through 65,535 (this is a subset of the 4-byte range).

Figure 14 on page 115 shows the sample topology.

Figure 14: Topology for Configuring the Local AS



In this example, Device R2 formerly belonged to AS 250 and now is in AS 200. Device R1 and Device R3 are configured to peer with AS 250 instead of with the new AS number (AS 200). Device R2 has the new AS number configured with the **autonomous-system 200** statement. To enable the peering sessions to work, the **local-as 250** statement is added in the BGP configuration. Because **local-as 250** is configured, Device R2 includes both the global AS (200) and the local AS (250) in its BGP inbound and outbound updates.

Configuration

- Configuring Device R1 on page 116
- Configuring Device R2 on page 118
- Configuring Device R3 on page 120

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 192.168.0.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext peer-as 250
set protocols bgp group ext neighbor 10.0.0.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 10.1.0.0/30 next-hop 10.0.0.2
set routing-options autonomous-system 100
```

Device R2

```
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 3 family inet address 10.1.0.1/30
set interfaces lo0 unit 2 family inet address 192.168.0.2/32
```

```
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext local-as 250
set protocols bgp group ext neighbor 10.0.0.1 peer-as 100
set protocols bgp group ext neighbor 10.1.0.2 peer-as 300
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options autonomous-system 200
```

Device R3

```
set interfaces fe-1/2/0 unit 4 family inet address 10.1.0.2/30
set interfaces lo0 unit 3 family inet address 192.168.0.3/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext peer-as 250
set protocols bgp group ext neighbor 10.1.0.1
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 10.0.0.0/30 next-hop 10.1.0.1
set routing-options autonomous-system 300
```

Configuring Device R1

Step-by-Step Procedure

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

To configure Device R1:

1. Configure the interfaces.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30


user@R1# set lo0 unit 1 family inet address 192.168.0.1/32
```

2. Configure external BGP (EBGP).

```
[edit protocols bgp group ext]
user@R1# set type external
user@R1# set export send-direct
user@R1# set export send-static
user@R1# set peer-as 250
user@R1# set neighbor 10.0.0.2
```

3. Configure the routing policy.

```
[edit policy-options]
user@R1# set policy-statement send-direct term 1 from protocol direct
user@R1# set policy-statement send-direct term 1 then accept
user@R1# set policy-statement send-static term 1 from protocol static
user@R1# set policy-statement send-static term 1 then accept
```

4. Configure a static route to the remote network between Device R2 and Device R3.

```
[edit routing-options]
user@R1# set static route 10.1.0.0/30 next-hop 10.0.0.2
```

5. Configure the global AS number.

```
[edit routing-options]
user@R1# set autonomous-system 100
```

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}
```

```
user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
policy-statement send-static {
  term 1 {
    from protocol static;
    then accept;
  }
}
```

```
user@R1# show protocols
bgp {
  group ext {
    type external;
    export [ send-direct send-static ];
    peer-as 250;
    neighbor 10.0.0.2;
  }
}
```

```
user@R1# show routing-options
static {
```

```
    route 10.1.0.0/30 next-hop 10.0.0.2;
}
autonomous-system 100;
```

When you are done configuring the device, enter **commit** from configuration mode.

Configuring Device R2

Step-by-Step Procedure

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

To configure Device R2:

1. Configure the interfaces.

```
[edit interfaces]
user@R2# set fe-1/2/0 unit 2 family inet address 10.0.0.2/30

user@R2# set fe-1/2/1 unit 3 family inet address 10.1.0.1/30

user@R2# set lo0 unit 2 family inet address 192.168.0.2/32
```

2. Configure EBGP.

```
[edit protocols bgp group ext]
user@R2# set type external
user@R2# set export send-direct
user@R2# set export send-static
user@R2# set neighbor 10.0.0.1 peer-as 100
user@R2# set neighbor 10.1.0.2 peer-as 300
```

3. Configure the local autonomous system (AS) number.

```
[edit protocols bgp group ext]
user@R2# set local-as 250
```

4. Configure the global AS number.

```
[edit routing-options]
user@R2# set autonomous-system 200
```

5. Configure the routing policy.

```
[edit policy-options]
user@R2# set policy-statement send-direct term 1 from protocol direct
user@R2# set policy-statement send-direct term 1 then accept
user@R2# set policy-statement send-static term 1 from protocol static
user@R2# set policy-statement send-static term 1 then accept
```

Results

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

```
user@R2# show interfaces
fe-1/2/0 {
    unit 2 {
```

```
      family inet {
        address 10.0.0.2/30;
      }
    }
}
fe-1/2/1 {
    unit 3 {
      family inet {
        address 10.1.0.1/30;
      }
    }
}
lo0 {
    unit 2 {
      family inet {
        address 192.168.0.2/32;
      }
    }
}
```

```
user@R2# show policy-options
policy-statement send-direct {
    term 1 {
      from protocol direct;
      then accept;
    }
}
policy-statement send-static {
    term 1 {
      from protocol static;
      then accept;
    }
}
```

```
user@R2# show protocols
bgp {
    group ext {
      type external;
      export [ send-direct send-static ];
      local-as 250;
      neighbor 10.0.0.1 {
        peer-as 100;
      }
      neighbor 10.1.0.2 {
        peer-as 300;
      }
    }
}
```

```
user@R2# show routing-options
autonomous-system 200;
```

When you are done configuring the device, enter **commit** from configuration mode.

Configuring Device R3

Step-by-Step Procedure

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

To configure Device R3:

1. Configure the interfaces.

```
[edit interfaces]
user@R3# set fe-1/2/0 unit 4 family inet address 10.1.0.2/30

user@R3# set lo0 unit 3 family inet address 192.168.0.3/32
```

2. Configure EBGP.

```
[edit protocols bgp group ext]
user@R3# set type external
user@R3# set export send-direct
user@R3# set export send-static
user@R3# set peer-as 250
user@R3# set neighbor 10.1.0.1
```

3. Configure the global autonomous system (AS) number.

```
[edit routing-options]
user@R3# set autonomous-system 300
```

4. Configure a static route to the remote network between Device R1 and Device R2.

```
[edit routing-options]
user@R3# set static route 10.0.0.0/30 next-hop 10.1.0.1
```

5. Configure the routing policy.

```
[edit policy-options]
user@R3# set policy-statement send-direct term 1 from protocol direct
user@R3# set policy-statement send-direct term 1 then accept
user@R3# set policy-statement send-static term 1 from protocol static
user@R3# set policy-statement send-static term 1 then accept
```

Results

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

```
user@R3# show interfaces
fe-1/2/0 {
   unit 4 {
      family inet {
         address 10.1.0.2/30;
      }
   }
}
lo0 {
   unit 3 {
```

```
      family inet {
        address 192.168.0.3/32;
      }
    }
  }
```

```
user@R3# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
policy-statement send-static {
  term 1 {
    from protocol static;
    then accept;
  }
}
```

```
user@R3# show protocols
bgp {
  group ext {
    type external;
    export [ send-direct send-static ];
    peer-as 250;
    neighbor 10.1.0.1;
  }
}
```

```
user@R3# show routing-options
static {
  route 10.0.0.0/30 next-hop 10.1.0.1;
}
autonomous-system 300;
```

When you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Checking the Local and Global AS Settings on page 121
- Checking the BGP Peering Sessions on page 123
- Verifying the BGP AS Paths on page 123

Checking the Local and Global AS Settings

Purpose Make sure that Device R2 has the local and global AS settings configured.

Action From operational mode, enter the **show bgp neighbors** command.

```
user@R2> show bgp neighbors
Peer: 10.0.0.1+179 AS 100      Local: 10.0.0.2+61036 AS 250
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
```

```
  Export: [ send-direct send-static ]
  Options: <Preference PeerAS LocalAS Refresh>
  Holdtime: 90 Preference: 170 Local AS: 250 Local System AS: 200
  Number of flaps: 0
  Peer ID: 192.168.0.1     Local ID: 192.168.0.2       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  Local Interface: fe-1/2/0.2
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 100)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              1
    Received prefixes:            3
    Accepted prefixes:            2
    Suppressed due to damping:    0
    Advertised prefixes:          4
  Last traffic (seconds): Received 6    Sent 14   Checked 47
  Input messages:  Total 258    Updates 3       Refreshes 0     Octets 4969
  Output messages: Total 258    Updates 2       Refreshes 0     Octets 5037
  Output Queue[0]: 0

Peer: 10.1.0.2+179 AS 300      Local: 10.1.0.1+52296 AS 250
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-direct send-static ]
  Options: <Preference PeerAS LocalAS Refresh>
  Holdtime: 90 Preference: 170 Local AS: 250 Local System AS: 200
  Number of flaps: 0
  Peer ID: 192.168.0.3     Local ID: 192.168.0.2       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  Local Interface: fe-1/2/1.3
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 300)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              1
    Received prefixes:            3
    Accepted prefixes:            2
```

```
  Suppressed due to damping:    0
  Advertised prefixes:          4
Last traffic (seconds): Received 19   Sent 26   Checked 9
Input messages:  Total 256    Updates 3       Refreshes 0     Octets 4931
Output messages: Total 256    Updates 2       Refreshes 0     Octets 4999
Output Queue[0]: 0
```

Meaning The Local AS: 250 and Local System AS: 200 output shows that Device R2 has the expected settings. Additionally, the output shows that the options list includes LocalAS.

Checking the BGP Peering Sessions

Purpose Ensure that the sessions are established and that the local AS number 250 is displayed.

Action From operational mode, enter the **show bgp summary** command.

```
user@R1> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 4          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.0.0.2                250        232        233       0       4     1:42:37
2/4/4/0              0/0/0/0
```

```
user@R3> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 4          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.1.0.1                250        235        236       0       4     1:44:25
2/4/4/0              0/0/0/0
```

Meaning Device R1 and Device R3 appear to be peering with a device in AS 250, even though Device R2 is actually in AS 200.

Verifying the BGP AS Paths

Purpose Make sure that the routes are in the routing tables and that the AS paths show the local AS number 250.

Action From configuration mode, enter the **set route protocol bgp** command.

```
user@R1> show route protocol bgp
inet.0: 6 destinations, 8 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30         [BGP/170] 01:46:44, localpref 100
                      AS path: 250 I
                    > to 10.0.0.2 via fe-1/2/0.1
10.1.0.0/30         [BGP/170] 01:46:44, localpref 100
                      AS path: 250 I
                    > to 10.0.0.2 via fe-1/2/0.1
192.168.0.2/32     *[BGP/170] 01:46:44, localpref 100
                      AS path: 250 I
                    > to 10.0.0.2 via fe-1/2/0.1
```

```
192.168.0.3/32     *[BGP/170] 01:46:40, localpref 100
                      AS path: 250 300 I
                    > to 10.0.0.2 via fe-1/2/0.1

user@R3> show route protocol bgp

inet.0: 6 destinations, 8 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/30         [BGP/170] 01:47:10, localpref 100
                      AS path: 250 I
                    > to 10.1.0.1 via fe-1/2/0.4
10.1.0.0/30         [BGP/170] 01:47:10, localpref 100
                      AS path: 250 I
                    > to 10.1.0.1 via fe-1/2/0.4
192.168.0.1/32     *[BGP/170] 01:47:10, localpref 100
                      AS path: 250 100 I
                    > to 10.1.0.1 via fe-1/2/0.4
192.168.0.2/32     *[BGP/170] 01:47:10, localpref 100
                      AS path: 250 I
                    > to 10.1.0.1 via fe-1/2/0.4
```

Meaning The output shows that Device R1 and Device R3 appear to have routes with AS paths that include AS 250, even though Device R2 is actually in AS 200.

Example: Configuring a Private Local AS for EBGP Sessions

This example shows how to configure a private local autonomous system (AS) number. The local AS is considered to be private because it is advertised to peers that use the local AS number for peering, but is hidden in the announcements to peers that can use the global AS number for peering.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

Use the **local-as** statement when ISPs merge and want to preserve a customer's configuration, particularly the AS with which the customer is configured to establish a peer relationship. The **local-as** statement simulates the AS number already in place in customer routers, even if the ISP's router has moved to a different AS.

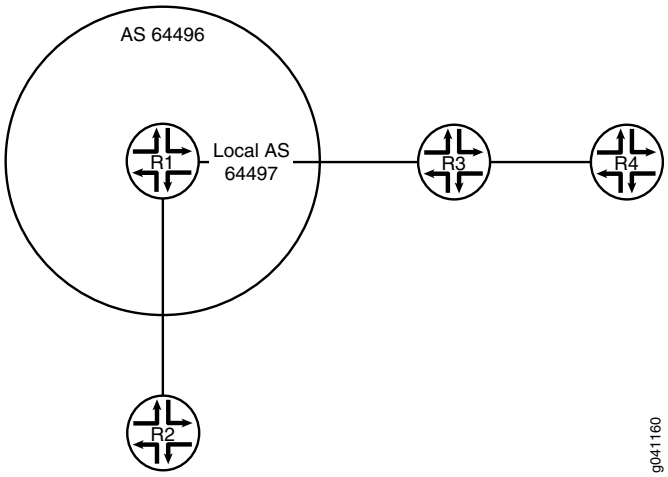
When you use the **private** option, the local AS is used during the establishment of the BGP session with an external BGP (EBGP) neighbor, but is hidden in the AS path sent to other EBGP peers. Only the global AS is included in the AS path sent to external peers.

The **private** option is useful for establishing local peering with routing devices that remain configured with their former AS or with a specific customer that has not yet modified its peer arrangements. The local AS is used to establish the BGP session with the EBGP neighbor, but is hidden in the AS path sent to external peers in another AS.

Include the **private** option so that the local AS is not prepended before the global AS in the AS path sent to external peers. When you specify the **private** option, the local AS is prepended only in the AS path sent to the EBGP neighbor.

Figure 15 on page 125 shows the sample topology.

Figure 15: Topology for Configuring a Private Local AS



Device R1 is in AS 64496. Device R2 is in AS 64510. Device R3 is in AS 64511. Device R4 is in AS 64512. Device R1 formerly belonged to AS 64497, which has merged with another network and now belongs to AS 64496. Because Device R3 still peers with Device R1, using its former AS, 64497, Device R1 needs to be configured with a local AS of 64497 in order to maintain peering with Device R3. Configuring a local AS of 64497 permits Device R1 to add AS 64497 when advertising routes to Device R3. Device R3 sees an AS path of 64497 64496 for the prefix 10.1.1.2/32, which is Device R2's loopback interface. Device R4, which is behind Device R3, sees an AS path of 64511 64497 64496 64510 to Device R2's loopback interface. To prevent Device R1 from adding the local AS number in its announcements to other peers, this example includes the **local-as 64497 private** statement. The **private** option configures Device R1 to not include the local AS 64497 when announcing routes to Device R2. Device R2 sees an AS path of 64496 64511 to Device R3 and an AS path of 64496 64511 64512 to Device R4. The **private** option in Device R1's configuration causes the AS number 64497 to be missing from the AS paths that Device R1 readvertises to Device R2.

Device R2 is hiding the private local AS from all the routers, except Device R3. The **private** option applies to the routes that Device R1 receives (learns) from Device R3 and that Device R1, in turn, readvertises to other routers. When these routes learned from Device R3 are readavertised by Device R1 to Device R2, the private local AS is missing from the AS path advertised to Device R2.

Configuration

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 3 family inet address 192.168.1.1/24
set interfaces fe-1/2/1 unit 5 family inet address 192.168.10.1/24
set interfaces lo0 unit 2 family inet address 10.1.1.1/32
set protocols bgp group external-AS64511 type external
set protocols bgp group external-AS64511 peer-as 64511
set protocols bgp group external-AS64511 local-as 64497
set protocols bgp group external-AS64511 local-as private
set protocols bgp group external-AS64511 neighbor 192.168.1.2
set protocols bgp group external-AS64510 type external
set protocols bgp group external-AS64510 peer-as 64510
set protocols bgp group external-AS64510 neighbor 192.168.10.2
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 64496
```

Device R2

```
set interfaces fe-1/2/0 unit 6 family inet address 192.168.10.2/24
set interfaces lo0 unit 3 family inet address 10.1.1.2/32
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 64496
set protocols bgp group external neighbor 192.168.10.1
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 64510
```

Device R3

```
set interfaces fe-1/2/0 unit 4 family inet address 192.168.1.2/24
set interfaces fe-1/2/1 unit 7 family inet address 192.168.5.1/24
set interfaces lo0 unit 4 family inet address 10.1.1.3/32
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external neighbor 192.168.1.1 peer-as 64497
set protocols bgp group external neighbor 192.168.5.2 peer-as 64512
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 64511
```

Device R4

```
set interfaces fe-1/2/0 unit 8 family inet address 192.168.5.2/24
set interfaces lo0 unit 5 family inet address 10.1.1.4/32
set protocols bgp group external type external
set protocols bgp group external export send-direct
set protocols bgp group external peer-as 64511
set protocols bgp group external neighbor 192.168.5.1
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 64512
```

Configuring Device R1

Step-by-Step Procedure

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

To configure Device R1:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 3]
user@R1# set family inet address 192.168.1.1/24

[edit interfaces fe-1/2/1 unit 5]
user@R1# set family inet address 192.168.10.1/24

[edit interfaces lo0 unit 2]
user@R1# set family inet address 10.1.1.1/32
```

2. Configure the EBGP peering session with Device R2.

```
[edit protocols bgp group external-AS64510]
user@R1# set type external
user@R1# set peer-as 64510
user@R1# set neighbor 192.168.10.2
```

3. Configure the EBGP peering session with Device R3.

```
[edit protocols bgp group external-AS64511]
user@R1# set type external
user@R1# set peer-as 64511
user@R1# set local-as 64497
user@R1# set local-as private
user@R1# set neighbor 192.168.1.2
```

4. Configure the routing policy.

```
[edit policy-options policy-statement send-direct term 1]
user@R1# set from protocol direct
user@R1# set then accept
```

5. Configure the global autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 64496
```

Results

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

```
user@R1# show interfaces
fe-1/2/0 {
   unit 3 {
      family inet {
         address 192.168.1.1/24;
      }
```

```
    }
  }
  fe-1/2/1 {
    unit 5 {
      family inet {
        address 192.168.10.1/24;
      }
    }
  }
  lo0 {
    unit 2 {
      family inet {
        address 10.1.1.1/32;
      }
    }
  }
```

```
user@R1# show policy-options
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
```

```
user@R1# show protocols
bgp {
  group external-AS64511 {
    type external;
    peer-as 64511;
    local-as 64497 private;
    neighbor 192.168.1.2;
  }
  group external-AS64510 {
    type external;
    peer-as 64510;
    neighbor 192.168.10.2;
  }
}
```

```
user@R1# show routing-options
autonomous-system 64496;
```

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

Repeat the configuration as needed for the other devices in the topology.

Verification

Confirm that the configuration is working properly.

- Checking Device R2's AS Paths on page 129
- Checking Device R3's AS Paths on page 129

Checking Device R2's AS Paths

Purpose Make sure that Device R2 does not have AS 64497 in its AS paths to Device R3 and Device R4.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@R2> show route protocol bgp
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.3/32        *[BGP/170] 01:33:11, localpref 100
                      AS path: 64496 64511 I
                    > to 192.168.10.1 via fe-1/2/0.6
10.1.1.4/32        *[BGP/170] 01:33:11, localpref 100
                      AS path: 64496 64511 64512 I
                    > to 192.168.10.1 via fe-1/2/0.6
192.168.5.0/24     *[BGP/170] 01:49:15, localpref 100
                      AS path: 64496 64511 I
                    > to 192.168.10.1 via fe-1/2/0.6
```

Meaning Device R2's AS paths do not include AS 64497.

Checking Device R3's AS Paths

Purpose Make sure that Device R3 does not have AS 64497 in its AS path to Device R4.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@R3> show route protocol bgp
inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.2/32        *[BGP/170] 01:35:11, localpref 100
                      AS path: 64497 64496 64510 I
                    > to 192.168.1.1 via fe-1/2/0.4
10.1.1.4/32        *[BGP/170] 01:35:11, localpref 100
                      AS path: 64512 I
                    > to 192.168.5.2 via fe-1/2/1.7
192.168.5.0/24      [BGP/170] 01:51:15, localpref 100
                      AS path: 64512 I
                    > to 192.168.5.2 via fe-1/2/1.7
```

Meaning Device R3's route to Device R2 (prefix 10.1.1.2) includes both the local and the global AS configured on Device R1 (64497 and 64496, respectively).

Related Documentation

- Examples: Configuring External BGP Peering on page 11
- *BGP Configuration Overview*

Example: Configuring the Accumulated IGP Attribute for BGP

- Understanding the Accumulated IGP Attribute for BGP on page 130
- Example: Configuring the Accumulated IGP Attribute for BGP on page 130

Understanding the Accumulated IGP Attribute for BGP

The interior gateway protocols (IGPs) are designed to handle routing within a single domain or an autonomous system (AS). Each link is assigned a particular value called a metric. The distance between the two nodes is calculated as a sum of all the metric values of links along the path. The IGP selects the shortest path between two nodes based on distance.

BGP is designed to provide routing over a large number of independent ASs with limited or no coordination among respective administrations. BGP does not use metrics in the path selection decisions.

The accumulated IGP (AIGP) metric attribute for BGP enables deployment in which a single administration can run several contiguous BGP ASs. Such deployments allow BGP to make routing decisions based on the IGP metric. In such networks, it is possible for BGP to select paths based on metrics as is done by IGPs. In this case, BGP chooses the shortest path between two nodes, even though the nodes might be in two different ASs.

The AIGP attribute is particularly useful in networks that use tunneling to deliver a packet to its BGP next hop. The Juniper Networks® Junos® operating system (Junos OS) currently supports the AIGP attribute for two BGP address families, **family inet labeled-unicast** and **family inet6 labeled-unicast**.

AIGP impacts the BGP best-route decision process. The AIGP attribute preference rule is applied after the local-preference rule. The AIGP distance is compared to break a tie. The BGP best-route decision process also impacts the way the interior cost rule is applied if the resolving next hop has an AIGP attribute. Without AIGP enabled, the interior cost of a route is based on the calculation of the metric to the next hop for the route. With AIGP enabled, the resolving AIGP distance is added to the interior cost.

The AIGP attribute is an optional non-transitive BGP path attribute and is specified in Internet draft draft-ietf-idr-aigp-06, *The Accumulated IGP Metric Attribute for BGP*.

Example: Configuring the Accumulated IGP Attribute for BGP

This example shows how to configure the accumulated IGP (AIGP) metric attribute for BGP.

Requirements

This example uses the following hardware and software components:

- Seven BGP-speaking devices.
- Junos OS Release 12.1 or later.

Overview

The AIGP attribute enables deployments in which a single administration can run several contiguous BGP autonomous systems (ASs). Such deployments allow BGP to make routing decisions based on the IGP metric. With AIGP enabled, BGP can select paths based on IGP metrics. This enables BGP to choose the shortest path between two nodes, even though the nodes might be in different ASs. The AIGP attribute is particularly useful in networks that use tunneling to deliver a packet to its BGP next hop. This example shows AIGP configured with MPLS label-switched paths.

To enable AIGP, you include the **aigp** statement in the BGP configuration on a protocol family basis. Configuring AIGP on a particular family enables sending and receiving of the AIGP attribute on that family. By default, AIGP is disabled. An AIGP-disabled neighbor does not send an AIGP attribute and silently discards a received AIGP attribute.

Junos OS supports AIGP for **family inet labeled-unicast** and **family inet6 labeled-unicast**. The **aigp** statement can be configured for a given family at the global BGP, group, or neighbor level.

By default, the value of the AIGP attribute for a local prefix is zero. An AIGP-enabled neighbor can originate an AIGP attribute for a given prefix by export policy, using the **aigp-originate** policy action. The value of the AIGP attribute reflects the IGP distance to the prefix. Alternatively, you can specify a value, by using the **aigp-originate distance *distance*** policy action. The configurable range is 0 through 4,294,967,295. Only one node needs to originate an AIGP attribute. The AIGP attribute is retained and readvertised if the neighbors are AIGP enabled with the **aigp** statement in the BGP configuration.

The policy action to originate the AIGP attribute has the following requirements:

- Neighbor must be AIGP enabled.
- Policy must be applied as an export policy.
- Prefix must have no current AIGP attribute.
- Prefix must export with next-hop self.
- Prefix must reside within the AIGP domain. Typically, a loopback IP address is the prefix to originate.

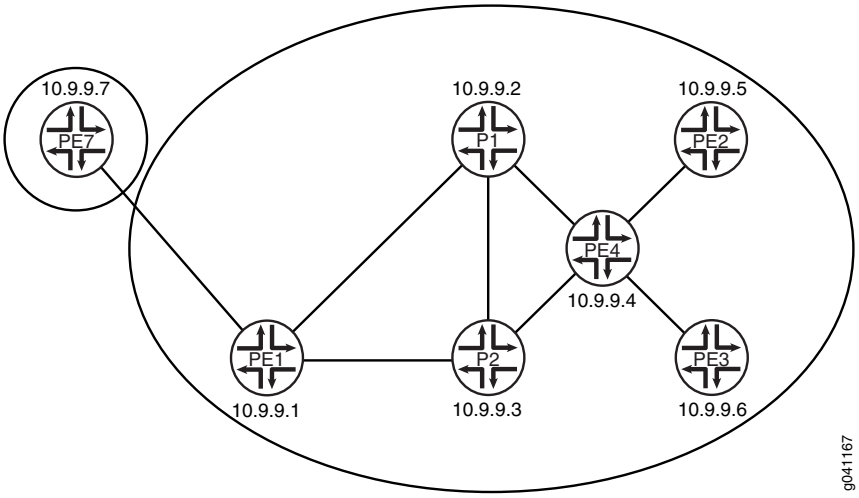
The policy is ignored if these requirements are not met.

Topology Diagram

Figure 16 on page 132 shows the topology used in this example. OSPF is used as the interior gateway protocol (IGP). Internal BGP (IBGP) is configured between Device PE1 and Device PE4. External BGP (EBGP) is configured between Device PE7 and Device PE1, between Device PE4 and Device PE3, and between Device PE4 and Device PE2. Devices PE4, PE2, and PE3 are configured for multihop. Device PE4 selects a path based on the AIGP value and then readvertises the AIGP value based on the AIGP and policy configuration. Device PE1 readvertises the AIGP value to Device PE7, which is in another administrative domain. Every device has two loopback interface addresses: 10.9.9.x is used for BGP peering and the router ID, and 10.100.1.x is used for the BGP next hop.

The network between Device PE1 and PE3 has IBGP peering and multiple OSPF areas. The external link to Device PE7 is configured to show that the AIGP attribute is readvertised to a neighbor outside of the administrative domain, if that neighbor is AIGP enabled.

Figure 16: Advertisement of Multiple Paths in BGP



For origination of an AIGP attribute, the BGP next hop is required to be itself. If the BGP next hop remains unchanged, the received AIGP attribute is readvertised, as is, to another AIGP neighbor. If the next hop changes, the received AIGP attribute is readvertised with an increased value to another AIGP neighbor. The increase in value reflects the IGP distance to the previous BGP next hop. To demonstrate, this example uses loopback interface addresses for Device PE4's EBGP peering sessions with Device PE2 and Device PE3. Multihop is enabled on these sessions so that a recursive lookup is performed to determine the point-to-point interface. Because the next hop changes, the IGP distance is added to the AIGP distance.

Configuration

- Configuring Device P1 on page 138
- Configuring Device P2 on page 141
- Configuring Device PE4 on page 144
- Configuring Device PE1 on page 149
- Configuring Device PE2 on page 153
- Configuring Device PE3 on page 157
- Configuring Device PE7 on page 160

CLI Quick Configuration

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

Device P1

```
set interfaces fe-1/2/0 unit 1 description P1-to-PE1
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.2/30
set interfaces fe-1/2/0 unit 1 family mpls
```

```
set interfaces fe-1/2/1 unit 4 description P1-to-P2
set interfaces fe-1/2/1 unit 4 family inet address 10.0.0.29/30
set interfaces fe-1/2/1 unit 4 family mpls
set interfaces fe-1/2/2 unit 8 description P1-to-PE4
set interfaces fe-1/2/2 unit 8 family inet address 10.0.0.17/30
set interfaces fe-1/2/2 unit 8 family mpls
set interfaces lo0 unit 3 family inet address 10.9.9.2/32
set interfaces lo0 unit 3 family inet address 10.100.1.2/32
set protocols rsvp interface fe-1/2/0.1
set protocols rsvp interface fe-1/2/2.8
set protocols rsvp interface fe-1/2/1.4
set protocols mpls label-switched-path P1-to-P2 to 10.9.9.3
set protocols mpls label-switched-path P1-to-PE1 to 10.9.9.1
set protocols mpls label-switched-path P1-to-PE4 to 10.9.9.4
set protocols mpls interface fe-1/2/0.1
set protocols mpls interface fe-1/2/2.8
set protocols mpls interface fe-1/2/1.4
set protocols bgp group internal type internal
set protocols bgp group internal local-address 10.9.9.2
set protocols bgp group internal family inet labeled-unicast aigp
set protocols bgp group internal neighbor 10.9.9.1
set protocols bgp group internal neighbor 10.9.9.3
set protocols bgp group internal neighbor 10.9.9.4
set protocols ospf area 0.0.0.1 interface fe-1/2/0.1 metric 1
set protocols ospf area 0.0.0.1 interface fe-1/2/1.4 metric 1
set protocols ospf area 0.0.0.0 interface fe-1/2/2.8 metric 1
set protocols ospf area 0.0.0.0 interface 10.9.9.2 passive
set protocols ospf area 0.0.0.0 interface 10.9.9.2 metric 1
set protocols ospf area 0.0.0.0 interface 10.100.1.2 passive
set protocols ospf area 0.0.0.0 interface 10.100.1.2 metric 1
set routing-options router-id 10.9.9.2
set routing-options autonomous-system 13979
```

Device P2

```
set interfaces fe-1/2/0 unit 3 description P2-to-PE1
set interfaces fe-1/2/0 unit 3 family inet address 10.0.0.6/30
set interfaces fe-1/2/0 unit 3 family mpls
set interfaces fe-1/2/1 unit 5 description P2-to-P1
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.30/30
set interfaces fe-1/2/1 unit 5 family mpls
set interfaces fe-1/2/2 unit 6 description P2-to-PE4
set interfaces fe-1/2/2 unit 6 family inet address 10.0.0.13/30
set interfaces fe-1/2/2 unit 6 family mpls
set interfaces lo0 unit 5 family inet address 10.9.9.3/32
set interfaces lo0 unit 5 family inet address 10.100.1.3/32
set protocols rsvp interface fe-1/2/1.5
set protocols rsvp interface fe-1/2/2.6
set protocols rsvp interface fe-1/2/0.3
set protocols mpls label-switched-path P2-to-PE1 to 10.9.9.1
set protocols mpls label-switched-path P2-to-P1 to 10.9.9.2
set protocols mpls label-switched-path P2-to-PE4 to 10.9.9.4
set protocols mpls interface fe-1/2/1.5
set protocols mpls interface fe-1/2/2.6
set protocols mpls interface fe-1/2/0.3
set protocols bgp group internal type internal
set protocols bgp group internal local-address 10.9.9.3
set protocols bgp group internal family inet labeled-unicast aigp
```

```
set protocols bgp group internal neighbor 10.9.9.1
set protocols bgp group internal neighbor 10.9.9.2
set protocols bgp group internal neighbor 10.9.9.4
set protocols ospf area 0.0.0.0 interface fe-1/2/2.6 metric 1
set protocols ospf area 0.0.0.0 interface 10.9.9.3 passive
set protocols ospf area 0.0.0.0 interface 10.9.9.3 metric 1
set protocols ospf area 0.0.0.0 interface 10.100.1.3 passive
set protocols ospf area 0.0.0.0 interface 10.100.1.3 metric 1
set routing-options router-id 10.9.9.3
set routing-options autonomous-system 13979
```

Device PE4

```
set interfaces fe-1/2/0 unit 7 description PE4-to-P2
set interfaces fe-1/2/0 unit 7 family inet address 10.0.0.14/30
set interfaces fe-1/2/0 unit 7 family mpls
set interfaces fe-1/2/1 unit 9 description PE4-to-P1
set interfaces fe-1/2/1 unit 9 family inet address 10.0.0.18/30
set interfaces fe-1/2/1 unit 9 family mpls
set interfaces fe-1/2/2 unit 10 description PE4-to-PE2
set interfaces fe-1/2/2 unit 10 family inet address 10.0.0.21/30
set interfaces fe-1/2/2 unit 10 family mpls
set interfaces fe-1/0/2 unit 12 description PE4-to-PE3
set interfaces fe-1/0/2 unit 12 family inet address 10.0.0.25/30
set interfaces fe-1/0/2 unit 12 family mpls
set interfaces lo0 unit 7 family inet address 10.9.9.4/32
set interfaces lo0 unit 7 family inet address 10.100.1.4/32
set protocols rsvp interface fe-1/2/0.7
set protocols rsvp interface fe-1/2/1.9
set protocols rsvp interface fe-1/2/2.10
set protocols rsvp interface fe-1/0/2.12
set protocols mpls label-switched-path PE4-to-PE2 to 10.9.9.5
set protocols mpls label-switched-path PE4-to-PE3 to 10.9.9.6
set protocols mpls label-switched-path PE4-to-P1 to 10.9.9.2
set protocols mpls label-switched-path PE4-to-P2 to 10.9.9.3
set protocols mpls interface fe-1/2/0.7
set protocols mpls interface fe-1/2/1.9
set protocols mpls interface fe-1/2/2.10
set protocols mpls interface fe-1/0/2.12
set protocols bgp export next-hop
set protocols bgp export aigp
set protocols bgp group internal type internal
set protocols bgp group internal local-address 10.9.9.4
set protocols bgp group internal family inet labeled-unicast aigp
set protocols bgp group internal neighbor 10.9.9.1
set protocols bgp group internal neighbor 10.9.9.3
set protocols bgp group internal neighbor 10.9.9.2
set protocols bgp group external type external
set protocols bgp group external multihop ttl 2
set protocols bgp group external local-address 10.9.9.4
set protocols bgp group external family inet labeled-unicast aigp
set protocols bgp group external peer-as 7018
set protocols bgp group external neighbor 10.9.9.5
set protocols bgp group external neighbor 10.9.9.6
set protocols ospf area 0.0.0.0 interface fe-1/2/1.9 metric 1
set protocols ospf area 0.0.0.0 interface fe-1/2/0.7 metric 1
set protocols ospf area 0.0.0.0 interface 10.9.9.4 passive
set protocols ospf area 0.0.0.0 interface 10.9.9.4 metric 1
```

```
set protocols ospf area 0.0.0.0 interface 10.100.1.4 passive
set protocols ospf area 0.0.0.0 interface 10.100.1.4 metric 1
set protocols ospf area 0.0.0.2 interface fe-1/2/2.10 metric 1
set protocols ospf area 0.0.0.3 interface fe-1/0/2.12 metric 1
set policy-options policy-statement aigp term 10 from protocol static
set policy-options policy-statement aigp term 10 from route-filter 44.0.0.0/24 exact
set policy-options policy-statement aigp term 10 then aigp-originate distance 200
set policy-options policy-statement aigp term 10 then next-hop 10.100.1.4
set policy-options policy-statement aigp term 10 then accept
set policy-options policy-statement next-hop term 10 from protocol bgp
set policy-options policy-statement next-hop term 10 then next-hop 10.100.1.4
set policy-options policy-statement next-hop term 10 then accept
set policy-options policy-statement next-hop term 20 from protocol direct
set policy-options policy-statement next-hop term 20 from route-filter 10.9.9.4/32 exact
set policy-options policy-statement next-hop term 20 from route-filter 10.100.1.4/32
  exact
set policy-options policy-statement next-hop term 20 then next-hop 10.100.1.4
set policy-options policy-statement next-hop term 20 then accept
set routing-options static route 44.0.0.0/24 discard
set routing-options router-id 10.9.9.4
set routing-options autonomous-system 13979
```

Device PE1

```
set interfaces fe-1/2/0 unit 0 description PE1-to-P1
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.1/30
set interfaces fe-1/2/0 unit 0 family mpls
set interfaces fe-1/2/1 unit 2 description PE1-to-P2
set interfaces fe-1/2/1 unit 2 family inet address 10.0.0.5/30
set interfaces fe-1/2/1 unit 2 family mpls
set interfaces fe-1/2/2 unit 14 description PE1-to-PE7
set interfaces fe-1/2/2 unit 14 family inet address 10.0.0.9/30
set interfaces lo0 unit 1 family inet address 10.9.9.1/32
set interfaces lo0 unit 1 family inet address 10.100.1.1/32
set protocols rsvp interface fe-1/2/0.0
set protocols rsvp interface fe-1/2/1.2
set protocols rsvp interface fe-1/2/2.14
set protocols mpls label-switched-path PE1-to-P1 to 10.9.9.2
set protocols mpls label-switched-path PE1-to-P2 to 10.9.9.3
set protocols mpls interface fe-1/2/0.0
set protocols mpls interface fe-1/2/1.2
set protocols mpls interface fe-1/2/2.14
set protocols bgp group internal type internal
set protocols bgp group internal local-address 10.9.9.1
set protocols bgp group internal family inet labeled-unicast aigp
set protocols bgp group internal export SET_EXPORT_ROUTES
set protocols bgp group internal vpn-apply-export
set protocols bgp group internal neighbor 10.9.9.4
set protocols bgp group internal neighbor 10.9.9.2
set protocols bgp group internal neighbor 10.9.9.3
set protocols bgp group external type external
set protocols bgp group external family inet labeled-unicast aigp
set protocols bgp group external export SET_EXPORT_ROUTES
set protocols bgp group external peer-as 7019
set protocols bgp group external neighbor 10.0.0.10
set protocols ospf area 0.0.0.1 interface fe-1/2/0.0 metric 1
set protocols ospf area 0.0.0.1 interface fe-1/2/1.2 metric 1
set protocols ospf area 0.0.0.1 interface 10.9.9.1 passive
```

```
set protocols ospf area 0.0.0.1 interface 10.9.9.1 metric 1
set protocols ospf area 0.0.0.1 interface 10.100.1.1 passive
set protocols ospf area 0.0.0.1 interface 10.100.1.1 metric 1
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol direct
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol bgp
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then next-hop
  10.100.1.1
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then accept
set routing-options router-id 10.9.9.1
set routing-options autonomous-system 13979
```

Device PE2

```
set interfaces fe-1/2/0 unit 11 description PE2-to-PE4
set interfaces fe-1/2/0 unit 11 family inet address 10.0.0.22/30
set interfaces fe-1/2/0 unit 11 family mpls
set interfaces lo0 unit 9 family inet address 10.9.9.5/32 primary
set interfaces lo0 unit 9 family inet address 10.100.1.5/32
set protocols rsvp interface fe-1/2/0.11
set protocols mpls label-switched-path PE2-to-PE4 to 10.9.9.4
set protocols mpls interface fe-1/2/0.11
set protocols bgp group external type external
set protocols bgp group external multihop ttl 2
set protocols bgp group external local-address 10.9.9.5
set protocols bgp group external family inet labeled-unicast aigp
set protocols bgp group external export next-hop
set protocols bgp group external export aigp
set protocols bgp group external export SET_EXPORT_ROUTES
set protocols bgp group external vpn-apply-export
set protocols bgp group external peer-as 13979
set protocols bgp group external neighbor 10.9.9.4
set protocols ospf area 0.0.0.2 interface 10.9.9.5 passive
set protocols ospf area 0.0.0.2 interface 10.9.9.5 metric 1
set protocols ospf area 0.0.0.2 interface 10.100.1.5 passive
set protocols ospf area 0.0.0.2 interface 10.100.1.5 metric 1
set protocols ospf area 0.0.0.2 interface fe-1/2/0.11 metric 1
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol direct
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol static
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol bgp
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then next-hop
  10.100.1.5
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then accept
set policy-options policy-statement aigp term 10 from route-filter 55.0.0.0/24 exact
set policy-options policy-statement aigp term 10 then aigp-originate distance 20
set policy-options policy-statement aigp term 10 then next-hop 10.100.1.5
set policy-options policy-statement aigp term 10 then accept
set policy-options policy-statement aigp term 20 from route-filter 99.0.0.0/24 exact
set policy-options policy-statement aigp term 20 then aigp-originate distance 30
set policy-options policy-statement aigp term 20 then next-hop 10.100.1.5
set policy-options policy-statement aigp term 20 then accept
set policy-options policy-statement next-hop term 10 from protocol bgp
set policy-options policy-statement next-hop term 10 then next-hop 10.100.1.5
set policy-options policy-statement next-hop term 10 then accept
set policy-options policy-statement next-hop term 20 from protocol direct
set policy-options policy-statement next-hop term 20 from route-filter 10.9.9.5/32 exact
set policy-options policy-statement next-hop term 20 from route-filter 10.100.1.5/32
  exact
set policy-options policy-statement next-hop term 20 then next-hop 10.100.1.5
```

```
set policy-options policy-statement next-hop term 20 then accept
set routing-options static route 99.0.0.0/24 discard
set routing-options static route 55.0.0.0/24 discard
set routing-options router-id 10.9.9.5
set routing-options autonomous-system 7018
```

Device PE3

```
set interfaces fe-1/2/0 unit 13 description PE3-to-PE4
set interfaces fe-1/2/0 unit 13 family inet address 10.0.0.26/30
set interfaces fe-1/2/0 unit 13 family mpls
set interfaces lo0 unit 11 family inet address 10.9.9.6/32
set interfaces lo0 unit 11 family inet address 10.100.1.6/32
set protocols rsvp interface fe-1/2/0.13
set protocols mpls label-switched-path PE3-to-PE4 to 10.9.9.4
set protocols mpls interface fe-1/2/0.13
set protocols bgp group external type external
set protocols bgp group external multihop ttl 2
set protocols bgp group external local-address 10.9.9.6
set protocols bgp group external family inet labeled-unicast aigp
set protocols bgp group external export next-hop
set protocols bgp group external export SET_EXPORT_ROUTES
set protocols bgp group external vpn-apply-export
set protocols bgp group external peer-as 13979
set protocols bgp group external neighbor 10.9.9.4
set protocols ospf area 0.0.0.3 interface 10.9.9.6 passive
set protocols ospf area 0.0.0.3 interface 10.9.9.6 metric 1
set protocols ospf area 0.0.0.3 interface 10.100.1.6 passive
set protocols ospf area 0.0.0.3 interface 10.100.1.6 metric 1
set protocols ospf area 0.0.0.3 interface fe-1/2/0.13 metric 1
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol direct
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol static
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol bgp
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then next-hop
  10.100.1.6
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then accept
set policy-options policy-statement next-hop term 10 from protocol bgp
set policy-options policy-statement next-hop term 10 then next-hop 10.100.1.6
set policy-options policy-statement next-hop term 10 then accept
set policy-options policy-statement next-hop term 20 from protocol direct
set policy-options policy-statement next-hop term 20 from route-filter 10.9.9.6/32 exact
set policy-options policy-statement next-hop term 20 from route-filter 10.100.1.6/32
  exact
set policy-options policy-statement next-hop term 20 then next-hop 10.100.1.6
set policy-options policy-statement next-hop term 20 then accept
set routing-options router-id 10.9.9.6
set routing-options autonomous-system 7018
```

Device PE7

```
set interfaces fe-1/2/0 unit 15 description PE7-to-PE1
set interfaces fe-1/2/0 unit 15 family inet address 10.0.0.10/30
set interfaces lo0 unit 13 family inet address 10.9.9.7/32
set interfaces lo0 unit 13 family inet address 10.100.1.7/32
set protocols bgp group external type external
set protocols bgp group external family inet labeled-unicast aigp
set protocols bgp group external export SET_EXPORT_ROUTES
set protocols bgp group external peer-as 13979
set protocols bgp group external neighbor 10.0.0.9
```

```
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol direct
set policy-options policy-statement SET_EXPORT_ROUTES term 10 from protocol bgp
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then next-hop
  10.100.1.7
set policy-options policy-statement SET_EXPORT_ROUTES term 10 then accept
set routing-options router-id 10.9.9.7
set routing-options autonomous-system 7019
```

Configuring Device P1

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 Device P1:

1. Configure the interfaces.

```
[edit interfaces]
user@P1# set fe-1/2/0 unit 1 description P1-to-PE1
user@P1# set fe-1/2/0 unit 1 family inet address 10.0.0.2/30
user@P1# set fe-1/2/0 unit 1 family mpls
user@P1# set fe-1/2/1 unit 4 description P1-to-P2
user@P1# set fe-1/2/1 unit 4 family inet address 10.0.0.29/30
user@P1# set fe-1/2/1 unit 4 family mpls
user@P1# set fe-1/2/2 unit 8 description P1-to-PE4
user@P1# set fe-1/2/2 unit 8 family inet address 10.0.0.17/30
user@P1# set fe-1/2/2 unit 8 family mpls
user@P1# set lo0 unit 3 family inet address 10.9.9.2/32
user@P1# set lo0 unit 3 family inet address 10.100.1.2/32
```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```
[edit protocols]
user@P1# set rsvp interface fe-1/2/0.1
user@P1# set rsvp interface fe-1/2/2.8
user@P1# set rsvp interface fe-1/2/1.4
user@P1# set mpls label-switched-path P1-to-P2 to 10.9.9.3
user@P1# set mpls label-switched-path P1-to-PE1 to 10.9.9.1
user@P1# set mpls label-switched-path P1-to-PE4 to 10.9.9.4
user@P1# set mpls interface fe-1/2/0.1
user@P1# set mpls interface fe-1/2/2.8
user@P1# set mpls interface fe-1/2/1.4
```

3. Configure BGP.

```
[edit protocols bgp group internal]
user@P1# set type internal
user@P1# set local-address 10.9.9.2
user@P1# set neighbor 10.9.9.1
user@P1# set neighbor 10.9.9.3
user@P1# set neighbor 10.9.9.4
```

4. Enable AIGP.

```
[edit protocols bgp group internal]
user@P1# set family inet labeled-unicast aigp
```

5. Configure an IGP, such as OSPF, RIP, or IS-IS.

```
[edit protocols ospf]
user@P1# set area 0.0.0.1 interface fe-1/2/0.1 metric 1
user@P1# set area 0.0.0.1 interface fe-1/2/1.4 metric 1
user@P1# set area 0.0.0.0 interface fe-1/2/2.8 metric 1
user@P1# set area 0.0.0.0 interface 10.9.9.2 passive
user@P1# set area 0.0.0.0 interface 10.9.9.2 metric 1
user@P1# set area 0.0.0.0 interface 10.100.1.2 passive
user@P1# set area 0.0.0.0 interface 10.100.1.2 metric 1
```

6. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@P1# set router-id 10.9.9.2
user@P1# set autonomous-system 13979
```

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

```
user@P1# commit
```

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

```
user@P1# show interfaces
fe-1/2/0 {
    unit 1 {
        description P1-to-PE1;
        family inet {
            address 10.0.0.2/30;
        }
        family mpls;
    }
}
fe-1/2/1 {
    unit 4 {
        description P1-to-P2;
        family inet {
            address 10.0.0.29/30;
        }
        family mpls;
    }
}
fe-1/2/2 {
    unit 8 {
        description P1-to-PE4;
        family inet {
            address 10.0.0.17/30;
        }
        family mpls;
    }
}
lo0 {
    unit 3 {
        family inet {
```

```
        address 10.9.9.2/32;
        address 10.100.1.2/32;
      }
    }
}
```

```
user@P1# show protocols
rsvp {
    interface fe-1/2/0.1;
    interface fe-1/2/2.8;
    interface fe-1/2/1.4;
}
mpls {
    label-switched-path P1-to-P2 {
        to 10.9.9.3;
    }
    label-switched-path P1-to-PE1 {
        to 10.9.9.1;
    }
    label-switched-path P1-to-PE4 {
        to 10.9.9.4;
    }
    interface fe-1/2/0.1;
    interface fe-1/2/2.8;
    interface fe-1/2/1.4;
}
bgp {
    group internal {
        type internal;
        local-address 10.9.9.2;
        family inet {
            labeled-unicast {
                aigp;
            }
        }
        neighbor 10.9.9.1;
        neighbor 10.9.9.3;
        neighbor 10.9.9.4;
    }
}
ospf {
    area 0.0.0.1 {
        interface fe-1/2/0.1 {
            metric 1;
        }
        interface fe-1/2/1.4 {
            metric 1;
        }
    }
    area 0.0.0.0 {
        interface fe-1/2/2.8 {
            metric 1;
        }
        interface 10.9.9.2 {
            passive;
            metric 1;
```

```
    }
    interface 10.100.1.2 {
      passive;
      metric 1;
    }
  }
}
```

```
user@P1# show routing-options
router-id 10.9.9.2;
autonomous-system 13979;
```

Configuring Device P2

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 Device P2:

1. Configure the interfaces.

```
[edit interfaces]
user@P2# set fe-1/2/0 unit 3 description P2-to-PE1
user@P2# set fe-1/2/0 unit 3 family inet address 10.0.0.6/30
user@P2# set fe-1/2/0 unit 3 family mpls
user@P2# set fe-1/2/1 unit 5 description P2-to-P1
user@P2# set fe-1/2/1 unit 5 family inet address 10.0.0.30/30
user@P2# set fe-1/2/1 unit 5 family mpls
user@P2# set fe-1/2/2 unit 6 description P2-to-PE4
user@P2# set fe-1/2/2 unit 6 family inet address 10.0.0.13/30
user@P2# set fe-1/2/2 unit 6 family mpls
user@P2# set lo0 unit 5 family inet address 10.9.9.3/32
user@P2# set lo0 unit 5 family inet address 10.100.1.3/32
```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```
[edit protocols]
user@P2# set rsvp interface fe-1/2/1.5
user@P2# set rsvp interface fe-1/2/2.6
user@P2# set rsvp interface fe-1/2/0.3
user@P2# set mpls label-switched-path P2-to-PE1 to 10.9.9.1
user@P2# set mpls label-switched-path P2-to-P1 to 10.9.9.2
user@P2# set mpls label-switched-path P2-to-PE4 to 10.9.9.4
user@P2# set mpls interface fe-1/2/1.5
user@P2# set mpls interface fe-1/2/2.6
user@P2# set mpls interface fe-1/2/0.3
```

3. Configure BGP.

```
[edit protocols bgp group internal]
user@P2# set type internal
user@P2# set local-address 10.9.9.3
user@P2# set neighbor 10.9.9.1
user@P2# set neighbor 10.9.9.2
user@P2# set neighbor 10.9.9.4
```

4. Enable AIGP.

```
[edit protocols bgp group internal]
user@P2# set family inet labeled-unicast aigp
```

5. Configure an IGP, such as OSPF, RIP, or IS-IS.

```
[edit protocols ospf]
user@P2# set area 0.0.0.0 interface fe-1/2/2.6 metric 1
user@P2# set area 0.0.0.0 interface 10.9.9.3 passive
user@P2# set area 0.0.0.0 interface 10.9.9.3 metric 1
user@P2# set area 0.0.0.0 interface 10.100.1.3 passive
user@P2# set area 0.0.0.0 interface 10.100.1.3 metric 1
```

6. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@P2# set router-id 10.9.9.3
user@P2# set autonomous-system 13979
```

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

```
user@P2# commit
```

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

```
user@P2# show interfaces
fe-1/2/0 {
    unit 3 {
        description P2-to-PE1;
        family inet {
            address 10.0.0.6/30;
        }
        family mpls;
    }
}
fe-1/2/1 {
    unit 5 {
        description P2-to-P1;
        family inet {
            address 10.0.0.30/30;
        }
        family mpls;
    }
}
fe-1/2/2 {
    unit 6 {
        description P2-to-PE4;
        family inet {
            address 10.0.0.13/30;
        }
        family mpls;
    }
}
lo0 {
```



```
    unit 5 {
      family inet {
        address 10.9.9.3/32;
        address 10.100.1.3/32;
      }
    }
}
user@P2# show protocols
rsvp {
    interface fe-1/2/1.5;
    interface fe-1/2/2.6;
    interface fe-1/2/0.3;
}
mpls {
    label-switched-path P2-to-PE1 {
      to 10.9.9.1;
    }
    label-switched-path P2-to-P1 {
      to 10.9.9.2;
    }
    label-switched-path P2-to-PE4 {
      to 10.9.9.4;
    }
    interface fe-1/2/1.5;
    interface fe-1/2/2.6;
    interface fe-1/2/0.3;
}
bgp {
    group internal {
      type internal;
      local-address 10.9.9.3;
      family inet {
        labeled-unicast {
          aigp;
        }
      }
      neighbor 10.9.9.1;
      neighbor 10.9.9.2;
      neighbor 10.9.9.4;
    }
}
ospf {
    area 0.0.0.0 {
      interface fe-1/2/2.6 {
        metric 1;
      }
      interface 10.9.9.3 {
        passive;
        metric 1;
      }
      interface 10.100.1.3 {
        passive;
        metric 1;
      }
    }
```

```
}

user@P2# show routing-options
router-id 10.9.9.3;
autonomous-system 13979;
```

Configuring Device PE4

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 Device PE4:

1. Configure the interfaces.

```
[edit interfaces]
user@PE4# set fe-1/2/0 unit 7 description PE4-to-P2
user@PE4# set fe-1/2/0 unit 7 family inet address 10.0.0.14/30
user@PE4# set fe-1/2/0 unit 7 family mpls
user@PE4# set fe-1/2/1 unit 9 description PE4-to-P1
user@PE4# set fe-1/2/1 unit 9 family inet address 10.0.0.18/30
user@PE4# set fe-1/2/1 unit 9 family mpls
user@PE4# set fe-1/2/2 unit 10 description PE4-to-PE2
user@PE4# set fe-1/2/2 unit 10 family inet address 10.0.0.21/30
user@PE4# set fe-1/2/2 unit 10 family mpls
user@PE4# set fe-1/0/2 unit 12 description PE4-to-PE3
user@PE4# set fe-1/0/2 unit 12 family inet address 10.0.0.25/30
user@PE4# set fe-1/0/2 unit 12 family mpls
user@PE4# set lo0 unit 7 family inet address 10.9.9.4/32
user@PE4# set lo0 unit 7 family inet address 10.100.1.4/32
```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```
[edit protocols]
user@PE4# set rsvp interface fe-1/2/0.7
user@PE4# set rsvp interface fe-1/2/1.9
user@PE4# set rsvp interface fe-1/2/2.10
user@PE4# set rsvp interface fe-1/0/2.12
user@PE4# set mpls label-switched-path PE4-to-PE2 to 10.9.9.5
user@PE4# set mpls label-switched-path PE4-to-PE3 to 10.9.9.6
user@PE4# set mpls label-switched-path PE4-to-P1 to 10.9.9.2
user@PE4# set mpls label-switched-path PE4-to-P2 to 10.9.9.3
user@PE4# set mpls interface fe-1/2/0.7
user@PE4# set mpls interface fe-1/2/1.9
user@PE4# set mpls interface fe-1/2/2.10
user@PE4# set mpls interface fe-1/0/2.12
```

3. Configure BGP.

```
[edit protocols bgp]
user@PE4# set export next-hop
user@PE4# set export aigp
user@PE4# set group internal type internal
user@PE4# set group internal local-address 10.9.9.4
user@PE4# set group internal neighbor 10.9.9.1
user@PE4# set group internal neighbor 10.9.9.3
```

```
user@PE4# set group internal neighbor 10.9.9.2
user@PE4# set group external type external
user@PE4# set group external multihop ttl 2
user@PE4# set group external local-address 10.9.9.4
user@PE4# set group external peer-as 7018
user@PE4# set group external neighbor 10.9.9.5
user@PE4# set group external neighbor 10.9.9.6
```

4. Enable AIGP.

```
[edit protocols bgp]
user@PE4# set group external family inet labeled-unicast aigp
user@PE4# set group internal family inet labeled-unicast aigp
```

5. Originate a prefix, and configure an AIGP distance.

   By default, a prefix is originated using the current IGP distance. Optionally, you can configure a distance for the AIGP attribute, using the **distance** option, as shown here.

```
[edit policy-options policy-statement aigp term 10]
user@PE4# set from protocol static
user@PE4# set from route-filter 44.0.0.0/24 exact
user@PE4# set then aigp-originate distance 200
user@PE4# set then next-hop 10.100.1.4
user@PE4# set then accept
```

6. Enable the policies.

```
[edit policy-options policy-statement next-hop]
user@PE4# set term 10 from protocol bgp
user@PE4# set term 10 then next-hop 10.100.1.4
user@PE4# set term 10 then accept
user@PE4# set term 20 from protocol direct
user@PE4# set term 20 from route-filter 10.9.9.4/32 exact
user@PE4# set term 20 from route-filter 10.100.1.4/32 exact
user@PE4# set term 20 then next-hop 10.100.1.4
user@PE4# set term 20 then accept
```

7. Configure a static route.

```
[edit routing-options]
user@PE4# set static route 44.0.0.0/24 discard
```

8. Configure an IGP, such as OSPF, RIP, or IS-IS.

```
[edit protocols ospf]
user@PE4# set area 0.0.0.0 interface fe-1/2/1.9 metric 1
user@PE4# set area 0.0.0.0 interface fe-1/2/0.7 metric 1
user@PE4# set area 0.0.0.0 interface 10.9.9.4 passive
user@PE4# set area 0.0.0.0 interface 10.9.9.4 metric 1
user@PE4# set area 0.0.0.0 interface 10.100.1.4 passive
user@PE4# set area 0.0.0.0 interface 10.100.1.4 metric 1
user@PE4# set area 0.0.0.2 interface fe-1/2/2.10 metric 1
user@PE4# set area 0.0.0.3 interface fe-1/0/2.12 metric 1
```

9. Configure the router ID and the autonomous system number.

```
[edit routing-options]
```

```
user@PE4# set router-id 10.9.9.4
user@PE4# set autonomous-system 13979
```

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

```
user@PE4# commit
```

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

```
user@PE4# show interfaces
fe-1/0/2 {
    unit 12 {
        description PE4-to-PE3;
        family inet {
            address 10.0.0.25/30;
        }
        family mpls;
    }
}
fe-1/2/0 {
    unit 7 {
        description PE4-to-P2;
        family inet {
            address 10.0.0.14/30;
        }
        family mpls;
    }
}
fe-1/2/1 {
    unit 9 {
        description PE4-to-P1;
        family inet {
            address 10.0.0.18/30;
        }
        family mpls;
    }
}
fe-1/2/2 {
    unit 10 {
        description PE4-to-PE2;
        family inet {
            address 10.0.0.21/30;
        }
        family mpls;
    }
}
lo0 {
    unit 7 {
        family inet {
            address 10.9.9.4/32;
            address 10.100.1.4/32;
        }
```

```
    }
}

user@PE4# show policy-options
policy-statement aigp {
    term 10 {
        from {
            protocol static;
            route-filter 44.0.0.0/24 exact;
        }
        then {
            aigp-originate distance 200;
            next-hop 10.100.1.4;
            accept;
        }
    }
}
policy-statement next-hop {
    term 10 {
        from protocol bgp;
        then {
            next-hop 10.100.1.4;
            accept;
        }
    }
    term 20 {
        from {
            protocol direct;
            route-filter 10.9.9.4/32 exact;
            route-filter 10.100.1.4/32 exact;
        }
        then {
            next-hop 10.100.1.4;
            accept;
        }
    }
}

user@PE4# show protocols
rsvp {
    interface fe-1/2/0.7;
    interface fe-1/2/1.9;
    interface fe-1/2/2.10;
    interface fe-1/0/2.12;
}
mpls {
    label-switched-path PE4-to-PE2 {
        to 10.9.9.5;
    }
    label-switched-path PE4-to-PE3 {
        to 10.9.9.6;
    }
    label-switched-path PE4-to-P1 {
        to 10.9.9.2;
    }
    label-switched-path PE4-to-P2 {
        to 10.9.9.3;
```

```
    }
    interface fe-1/2/0.7;
    interface fe-1/2/1.9;
    interface fe-1/2/2.10;
    interface fe-1/0/2.12;
}
bgp {
    export [ next-hop aigp ];
    group internal {
        type internal;
        local-address 10.9.9.4;
        family inet {
            labeled-unicast {
                aigp;
            }
        }
        neighbor 10.9.9.1;
        neighbor 10.9.9.3;
        neighbor 10.9.9.2;
    }
    group external {
        type external;
        multihop {
            ttl 2;
        }
        local-address 10.9.9.4;
        family inet {
            labeled-unicast {
                aigp;
            }
        }
        peer-as 7018;
        neighbor 10.9.9.5;
        neighbor 10.9.9.6;
    }
}
ospf {
    area 0.0.0.0 {
        interface fe-1/2/1.9 {
            metric 1;
        }
        interface fe-1/2/0.7 {
            metric 1;
        }
        interface 10.9.9.4 {
            passive;
            metric 1;
        }
        interface 10.100.1.4 {
            passive;
            metric 1;
        }
    }
    area 0.0.0.2 {
        interface fe-1/2/2.10 {
            metric 1;
```

```
    }
  }
  area 0.0.0.3 {
    interface fe-1/0/2.12 {
      metric 1;
    }
  }
}
```

```
user@PE4# show routing-options
static {
  route 44.0.0.0/24 discard;
}
router-id 10.9.9.4;
autonomous-system 13979;
```

Configuring Device PE1

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 Device PE1:

1. Configure the interfaces.

```
[edit interfaces]
user@PE1# set fe-1/2/0 unit 0 description PE1-to-P1
user@PE1# set fe-1/2/0 unit 0 family inet address 10.0.0.1/30
user@PE1# set fe-1/2/0 unit 0 family mpls
user@PE1# set fe-1/2/1 unit 2 description PE1-to-P2
user@PE1# set fe-1/2/1 unit 2 family inet address 10.0.0.5/30
user@PE1# set fe-1/2/1 unit 2 family mpls
user@PE1# set fe-1/2/2 unit 14 description PE1-to-PE7
user@PE1# set fe-1/2/2 unit 14 family inet address 10.0.0.9/30
user@PE1# set lo0 unit 1 family inet address 10.9.9.1/32
user@PE1# set lo0 unit 1 family inet address 10.100.1.1/32
```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```
[edit protocols]
user@PE1# set rsvp interface fe-1/2/0.0
user@PE1# set rsvp interface fe-1/2/1.2
user@PE1# set rsvp interface fe-1/2/2.14
user@PE1# set mpls label-switched-path PE1-to-P1 to 10.9.9.2
user@PE1# set mpls label-switched-path PE1-to-P2 to 10.9.9.3
user@PE1# set mpls interface fe-1/2/0.0
user@PE1# set mpls interface fe-1/2/1.2
user@PE1# set mpls interface fe-1/2/2.14
```

3. Configure BGP.

```
[edit protocols bgp]
user@PE1# set group internal type internal
user@PE1# set group internal local-address 10.9.9.1
user@PE1# set group internal export SET_EXPORT_ROUTES
user@PE1# set group internal vpn-apply-export
```

```
user@PE1# set group internal neighbor 10.9.9.4
user@PE1# set group internal neighbor 10.9.9.2
user@PE1# set group internal neighbor 10.9.9.3
user@PE1# set group external type external
user@PE1# set group external export SET_EXPORT_ROUTES
user@PE1# set group external peer-as 7019
user@PE1# set group external neighbor 10.0.0.10
```

4. Enable AIGP.

```
[edit protocols bgp]
user@PE1# set group internal family inet labeled-unicast aigp
user@PE1# set group external family inet labeled-unicast aigp
```

5. Enable the policies.

```
[edit policy-options policy-statement SET_EXPORT_ROUTES term 10]
user@PE1# set from protocol direct
user@PE1# set from protocol bgp
user@PE1# set then next-hop 10.100.1.1
user@PE1# set then accept
```

6. Configure an IGP, such as OSPF, RIP, or IS-IS.

```
[edit protocols ospf area 0.0.0.1]
user@PE1# set interface fe-1/2/0.0 metric 1
user@PE1# set interface fe-1/2/1.2 metric 1
user@PE1# set interface 10.9.9.1 passive
user@PE1# set interface 10.9.9.1 metric 1
user@PE1# set interface 10.100.1.1 passive
user@PE1# set interface 10.100.1.1 metric 1
```

7. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@PE1# set router-id 10.9.9.1
user@PE1# set autonomous-system 13979
```

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

```
user@PE1# commit
```

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

```
user@PE1# show interfaces
fe-1/2/0 {
  unit 0 {
    description PE1-to-P1;
    family inet {
      address 10.0.0.1/30;
    }
    family mpls;
  }
}
fe-1/2/1 {
```



```
    unit 2 {
      description PE1-to-P2;
      family inet {
        address 10.0.0.5/30;
      }
      family mpls;
    }
  }
  fe-1/2/2 {
    unit 14 {
      description PE1-to-PE7;
      family inet {
        address 10.0.0.9/30;
      }
    }
  }
  lo0 {
    unit 1 {
      family inet {
        address 10.9.9.1/32;
        address 10.100.1.1/32;
      }
    }
  }
```

```
user@PE1# show policy-options
policy-statement SET_EXPORT_ROUTES {
  term 10 {
    from protocol [ direct bgp ];
    then {
      next-hop 10.100.1.1;
      accept;
    }
  }
}
```

```
user@PE1# show protocols
rsvp {
  interface fe-1/2/0.0;
  interface fe-1/2/1.2;
  interface fe-1/2/2.14;
}
mpls {
  label-switched-path PE1-to-P1 {
    to 10.9.9.2;
  }
  label-switched-path PE1-to-P2 {
    to 10.9.9.3;
  }
  interface fe-1/2/0.0;
  interface fe-1/2/1.2;
  interface fe-1/2/2.14;
}
bgp {
  group internal {
    type internal;
    local-address 10.9.9.1;
```

```
      family inet {
        labeled-unicast {
          aigp;
        }
      }
      export SET_EXPORT_ROUTES;
      vpn-apply-export;
      neighbor 10.9.9.4;
      neighbor 10.9.9.2;
      neighbor 10.9.9.3;
    }
    group external {
      type external;
      family inet {
        labeled-unicast {
          aigp;
        }
      }
      export SET_EXPORT_ROUTES;
      peer-as 7019;
      neighbor 10.0.0.10;
    }
}
ospf {
    area 0.0.0.1 {
      interface fe-1/2/0.0 {
        metric 1;
      }
      interface fe-1/2/1.2 {
        metric 1;
      }
      interface 10.9.9.1 {
        passive;
        metric 1;
      }
      interface 10.100.1.1 {
        passive;
        metric 1;
      }
    }
}
```

```
user@PE1# show routing-options
router-id 10.9.9.1;
autonomous-system 13979;
```

Configuring Device PE2

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 Device PE2:

1. Configure the interfaces.

```
[edit interfaces]
user@PE2# set fe-1/2/0 unit 11 description PE2-to-PE4
user@PE2# set fe-1/2/0 unit 11 family inet address 10.0.0.22/30
user@PE2# set fe-1/2/0 unit 11 family mpls
user@PE2# set lo0 unit 9 family inet address 10.9.9.5/32 primary
user@PE2# set lo0 unit 9 family inet address 10.100.1.5/32
```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```
[edit protocols]
user@PE2# set rsvp interface fe-1/2/0.11
user@PE2# set mpls label-switched-path PE2-to-PE4 to 10.9.9.4
user@PE2# set mpls interface fe-1/2/0.11
```

3. Configure BGP.

```
[edit protocols bgp]
user@PE2# set group external type external
user@PE2# set group external multihop ttl 2
user@PE2# set group external local-address 10.9.9.5
user@PE2# set group external export next-hop
user@PE2# set group external export aigp
user@PE2# set group external export SET_EXPORT_ROUTES
user@PE2# set group external vpn-apply-export
user@PE2# set group external peer-as 13979
user@PE2# set group external neighbor 10.9.9.4
```

4. Enable AIGP.

```
[edit protocols bgp]
user@PE2# set group external family inet labeled-unicast aigp
```

5. Originate a prefix, and configure an AIGP distance.

   By default, a prefix is originated using the current IGP distance. Optionally, you can configure a distance for the AIGP attribute, using the **distance** option, as shown here.

```
[edit policy-options policy-statement aigp]
user@PE2# set term 10 from route-filter 55.0.0.0/24 exact
user@PE2# set term 10 then aigp-originate distance 20
user@PE2# set term 10 then next-hop 10.100.1.5
user@PE2# set term 10 then accept
user@PE2# set term 20 from route-filter 99.0.0.0/24 exact
user@PE2# set term 20 then aigp-originate distance 30
user@PE2# set term 20 then next-hop 10.100.1.5
user@PE2# set term 20 then accept
```

6. Enable the policies.

```
[edit policy-options]
user@PE2# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
  direct
user@PE2# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
  static
user@PE2# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
  bgp
user@PE2# set policy-statement SET_EXPORT_ROUTES term 10 then next-hop
  10.100.1.5
user@PE2# set policy-statement SET_EXPORT_ROUTES term 10 then accept
user@PE2# set policy-statement next-hop term 10 from protocol bgp
user@PE2# set policy-statement next-hop term 10 then next-hop 10.100.1.5
user@PE2# set policy-statement next-hop term 10 then accept
user@PE2# set policy-statement next-hop term 20 from protocol direct
user@PE2# set policy-statement next-hop term 20 from route-filter 10.9.9.5/32
  exact
user@PE2# set policy-statement next-hop term 20 from route-filter 10.100.1.5/32
  exact
user@PE2# set policy-statement next-hop term 20 then next-hop 10.100.1.5
user@PE2# set policy-statement next-hop term 20 then accept
```

7. Enable some static routes.

```
[edit routing-options]
user@PE2# set static route 99.0.0.0/24 discard
user@PE2# set static route 55.0.0.0/24 discard
```

8. Configure an IGP, such as OSPF, RIP, or IS-IS.

```
[edit protocols ospf area 0.0.0.2]
user@PE2# set interface 10.9.9.5 passive
user@PE2# set interface 10.9.9.5 metric 1
user@PE2# set interface 10.100.1.5 passive
user@PE2# set interface 10.100.1.5 metric 1
user@PE2# set interface fe-1/2/0.11 metric 1
```

9. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@PE2# set router-id 10.9.9.5
user@PE2# set autonomous-system 7018
```

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

```
user@PE2# commit
```

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

```
user@PE2# show interfaces
fe-1/2/0 {
  unit 11 {
    description PE2-to-PE4;
```



```
      family inet {
        address 10.0.0.22/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 9 {
      family inet {
        address 10.9.9.5/32 {
          primary;
        }
        address 10.100.1.5/32;
      }
    }
  }
```

```
user@PE2# show policy-options
policy-statement SET_EXPORT_ROUTES {
  term 10 {
    from protocol [ direct static bgp ];
    then {
      next-hop 10.100.1.5;
      accept;
    }
  }
}
policy-statement aigp {
  term 10 {
    from {
      route-filter 55.0.0.0/24 exact;
    }
    then {
      aigp-originate distance 20;
      next-hop 10.100.1.5;
      accept;
    }
  }
  term 20 {
    from {
      route-filter 99.0.0.0/24 exact;
    }
    then {
      aigp-originate distance 30;
      next-hop 10.100.1.5;
      accept;
    }
  }
}
policy-statement next-hop {
  term 10 {
    from protocol bgp;
    then {
      next-hop 10.100.1.5;
      accept;
    }
```

```
    }
    term 20 {
      from {
        protocol direct;
        route-filter 10.9.9.5/32 exact;
        route-filter 10.100.1.5/32 exact;
      }
      then {
        next-hop 10.100.1.5;
        accept;
      }
    }
}
user@PE2# show protocols
rsvp {
    interface fe-1/2/0.11;
}
mpls {
    label-switched-path PE2-to-PE4 {
        to 10.9.9.4;
    }
    interface fe-1/2/0.11;
}
bgp {
    group external {
        type external;
        multihop {
            ttl 2;
        }
        local-address 10.9.9.5;
        family inet {
            labeled-unicast {
                aigp;
            }
        }
        export [ next-hop aigp SET_EXPORT_ROUTES ];
        vpn-apply-export;
        peer-as 13979;
        neighbor 10.9.9.4;
    }
}
ospf {
    area 0.0.0.2 {
        interface 10.9.9.5 {
            passive;
            metric 1;
        }
        interface 10.100.1.5 {
            passive;
            metric 1;
        }
        interface fe-1/2/0.11 {
            metric 1;
        }
    }
```

```
}
user@PE2# show routing-options
static {
   route 99.0.0.0/24 discard;
   route 55.0.0.0/24 discard;
}
router-id 10.9.9.5;
autonomous-system 7018;
```

Configuring Device PE3

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 Device PE3:

1. Configure the interfaces.

```
[edit interfaces]
user@PE3# set fe-1/2/0 unit 13 description PE3-to-PE4
user@PE3# set fe-1/2/0 unit 13 family inet address 10.0.0.26/30
user@PE3# set fe-1/2/0 unit 13 family mpls
user@PE3# set lo0 unit 11 family inet address 10.9.9.6/32
user@PE3# set lo0 unit 11 family inet address 10.100.1.6/32
```

2. Configure MPLS and a signaling protocol, such as RSVP or LDP.

```
[edit protocols]
user@PE3# set rsvp interface fe-1/2/0.13
user@PE3# set mpls label-switched-path PE3-to-PE4 to 10.9.9.4
user@PE3# set mpls interface fe-1/2/0.13
```

3. Configure BGP.

```
[edit protocols bgp group external]
user@PE3# set type external
user@PE3# set multihop ttl 2
user@PE3# set local-address 10.9.9.6
user@PE3# set export next-hop
user@PE3# set export SET_EXPORT_ROUTES
user@PE3# set vpn-apply-export
user@PE3# set peer-as 13979
user@PE3# set neighbor 10.9.9.4
```

4. Enable AIGP.

```
[edit protocols bgp group external]
user@PE3# set family inet labeled-unicast aigp
```

5. Enable the policies.

```
[edit policy-options]
user@PE3# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
   direct
user@PE3# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
   static
```

```
user@PE3# set policy-statement SET_EXPORT_ROUTES term 10 from protocol
  bgp
user@PE3# set policy-statement SET_EXPORT_ROUTES term 10 then next-hop
  10.100.1.6
user@PE3# set policy-statement SET_EXPORT_ROUTES term 10 then accept
user@PE3# set policy-statement next-hop term 10 from protocol bgp
user@PE3# set policy-statement next-hop term 10 then next-hop 10.100.1.6
user@PE3# set policy-statement next-hop term 10 then accept
user@PE3# set policy-statement next-hop term 20 from protocol direct
user@PE3# set policy-statement next-hop term 20 from route-filter 10.9.9.6/32
  exact
user@PE3# set policy-statement next-hop term 20 from route-filter 10.100.1.6/32
  exact
user@PE3# set policy-statement next-hop term 20 then next-hop 10.100.1.6
user@PE3# set policy-statement next-hop term 20 then accept
```

6. Configure an IGP, such as OSPF, RIP, or IS-IS.

```
[edit protocols ospf area 0.0.0.3]
user@PE3# set interface 10.9.9.6 passive
user@PE3# set interface 10.9.9.6 metric 1
user@PE3# set interface 10.100.1.6 passive
user@PE3# set interface 10.100.1.6 metric 1
user@PE3# set interface fe-1/2/0.13 metric 1
```

7. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@PE3# set router-id 10.9.9.6
user@PE3# set autonomous-system 7018
```

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

```
user@PE3# commit
```

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

```
user@PE3# show interfaces
fe-1/2/0 {
   unit 13 {
      description PE3-to-PE4;
      family inet {
         address 10.0.0.26/30;
      }
      family mpls;
   }
}
lo0 {
   unit 11 {
      family inet {
         address 10.9.9.6/32;
         address 10.100.1.6/32;
      }
```

```
    }
}

user@PE3# show policy-options
policy-statement SET_EXPORT_ROUTES {
    term 10 {
        from protocol [ direct static bgp ];
        then {
            next-hop 10.100.1.6;
            accept;
        }
    }
}
policy-statement next-hop {
    term 10 {
        from protocol bgp;
        then {
            next-hop 10.100.1.6;
            accept;
        }
    }
    term 20 {
        from {
            protocol direct;
            route-filter 10.9.9.6/32 exact;
            route-filter 10.100.1.6/32 exact;
        }
        then {
            next-hop 10.100.1.6;
            accept;
        }
    }
}

user@PE3# show protocols
rsvp {
    interface fe-1/2/0.13;
}
mpls {
    label-switched-path PE3-to-PE4 {
        to 10.9.9.4;
    }
    interface fe-1/2/0.13;
}
bgp {
    group external {
        type external;
        multihop {
            ttl 2;
        }
        local-address 10.9.9.6;
        family inet {
            labeled-unicast {
                aigp;
            }
        }
        export [ next-hop SET_EXPORT_ROUTES ];
```

```
      vpn-apply-export;
      peer-as 13979;
      neighbor 10.9.9.4;
    }
  }
  ospf {
    area 0.0.0.3 {
      interface 10.9.9.6 {
        passive;
        metric 1;
      }
      interface 10.100.1.6 {
        passive;
        metric 1;
      }
      interface fe-1/2/0.13 {
        metric 1;
      }
    }
  }
```

```
user@PE3# show routing-options
router-id 10.9.9.6;
autonomous-system 7018;
```

Configuring Device PE7

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 Device PE7:

1. Configure the interfaces.

   ```
   [edit interfaces]
   user@PE7# set fe-1/2/0 unit 15 description PE7-to-PE1
   user@PE7# set fe-1/2/0 unit 15 family inet address 10.0.0.10/30
   user@PE7# set lo0 unit 13 family inet address 10.9.9.7/32
   user@PE7# set lo0 unit 13 family inet address 10.100.1.7/32
   ```

2. Configure BGP.

   ```
   [edit protocols bgp group external]
   user@PE7# set type external
   user@PE7# set export SET_EXPORT_ROUTES
   user@PE7# set peer-as 13979
   user@PE7# set neighbor 10.0.0.9
   ```

3. Enable AIGP.

   ```
   [edit protocols bgp group external]
   user@PE7# set family inet labeled-unicast aigp
   ```

4. Configure the routing policy.

   ```
   [edit policy-options policy-statement SET_EXPORT_ROUTES term 10]
   user@PE7# set from protocol direct
   ```

```
user@PE7# set from protocol bgp
user@PE7# set then next-hop 10.100.1.7
user@PE7# set then accept
```

5. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@PE7# set router-id 10.9.9.7
user@PE7# set autonomous-system 7019
```

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

```
user@PE7# commit
```

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

```
user@PE7# show interfaces
interfaces {
  fe-1/2/0 {
    unit 15 {
      description PE7-to-PE1;
      family inet {
        address 10.0.0.10/30;
      }
    }
  }
  lo0 {
    unit 13 {
      family inet {
        address 10.9.9.7/32;
        address 10.100.1.7/32;
      }
    }
  }
}
```

```
user@PE7# show policy-options
policy-statement SET_EXPORT_ROUTES {
  term 10 {
    from protocol [ direct bgp ];
    then {
      next-hop 10.100.1.7;
      accept;
    }
  }
}
```

```
user@PE7# show protocols
bgp {
  group external {
    type external;
    family inet {
      labeled-unicast {
        aigp;
```

```
        }
      }
      export SET_EXPORT_ROUTES;
      peer-as 13979;
      neighbor 10.0.0.9;
    }
  }
```

```
user@PE7# show routing-options
router-id 10.9.9.7;
autonomous-system 7019;
```

Verification

Confirm that the configuration is working properly.

- Verifying That Device PE4 Is Receiving the AIGP Attribute from Its EBGP Neighbor PE2 on page 162
- Checking the IGP Metric on page 162
- Verifying That Device PE4 Adds the IGP Metric to the AIGP Attribute on page 163
- Verifying That Device PE7 Is Receiving the AIGP Attribute from Its EBGP Neighbor PE1 on page 163
- Verifying the Resolving AIGP Metric on page 164
- Verifying the Presence of AIGP Attributes in BGP Updates on page 167

Verifying That Device PE4 Is Receiving the AIGP Attribute from Its EBGP Neighbor PE2

Purpose Make sure that the AIGP policy on Device PE2 is working.

Action

```
user@PE4> show route receive-protocol bgp 10.9.9.5 extensive
* 55.0.0.0/24 (1 entry, 1 announced)
     Accepted
     Route Label: 299888
     Nexthop: 10.100.1.5
     AS path: 7018 I
     AIGP: 20

* 99.0.0.0/24 (1 entry, 1 announced)
     Accepted
     Route Label: 299888
     Nexthop: 10.100.1.5
     AS path: 7018 I
     AIGP: 30
```

Meaning On Device PE2, the **aigp-originate** statement is configured with a distance of 20 (**aigp-originate distance 20**). This statement is applied to route 55.0.0.0/24. Likewise, the **aigp-originate distance 30** statement is applied to route 99.0.0.0/24. Thus, when Device PE4 receives these routes, the AIGP attribute is attached with the configured metrics.

Checking the IGP Metric

Purpose From Device PE4, check the IGP metric to the BGP next hop 10.100.1.5.

Action

```
user@PE4> show route 10.100.1.5
inet.0: 30 destinations, 40 routes (30 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.100.1.5/32      *[OSPF/10] 05:35:50, metric 2
                    > to 10.0.0.22 via fe-1/2/2.10
                    [BGP/170] 03:45:07, localpref 100, from 10.9.9.5
                      AS path: 7018 I
                    > to 10.0.0.22 via fe-1/2/2.10
```

Meaning The IGP metric for this route is 2.

Verifying That Device PE4 Adds the IGP Metric to the AIGP Attribute

Purpose Make sure that Device PE4 adds the IGP metric to the AIGP attribute when it readvertises routes to its IBGP neighbor, Device PE1.

Action

```
user@PE4> show route advertising-protocol bgp 10.9.9.1 extensive

* 55.0.0.0/24 (1 entry, 1 announced)
 BGP group internal type Internal
     Route Label: 300544
     Nexthop: 10.100.1.4
     Flags: Nexthop Change
     Localpref: 100
     AS path: [13979] 7018 I
     AIGP: 22

* 99.0.0.0/24 (1 entry, 1 announced)
 BGP group internal type Internal
     Route Label: 300544
     Nexthop: 10.100.1.4
     Flags: Nexthop Change
     Localpref: 100
     AS path: [13979] 7018 I
     AIGP: 32
```

Meaning The IGP metric is added to the AIGP metric (20 + 2 = 22 and 30 + 2 = 32), because the next hop is changed for these routes.

Verifying That Device PE7 Is Receiving the AIGP Attribute from Its EBGP Neighbor PE1

Purpose Make sure that the AIGP policy on Device PE1 is working.

Action

```
user@PE7> show route receive-protocol bgp 10.0.0.9 extensive
* 44.0.0.0/24 (1 entry, 1 announced)
     Accepted
     Route Label: 300096
     Nexthop: 10.0.0.9
     AS path: 13979 I
     AIGP: 203

* 55.0.0.0/24 (1 entry, 1 announced)
     Accepted
     Route Label: 300112
     Nexthop: 10.0.0.9
     AS path: 13979 7018 I
     AIGP: 25

* 99.0.0.0/24 (1 entry, 1 announced)
     Accepted
     Route Label: 300112
     Nexthop: 10.0.0.9
     AS path: 13979 7018 I
     AIGP: 35
```

Meaning The 44.0.0.0/24 route is originated at Device PE4. The 55.0.0.0/24 and 99.0.0.0/24 routes are originated at Device PE2. The IGP distances are added to the configured AIGP distances.

Verifying the Resolving AIGP Metric

Purpose Confirm that if the prefix is resolved through recursion and the recursive next hops have AIGP metrics, the prefix has the sum of the AIGP values that are on the recursive BGP next hops.

Action

1. Add a static route to 66.0.0.0/24.

   ```
   [edit routing-options]
   user@PE2# set static route 66.0.0.0/24 discard
   ```

2. Delete the existing terms in the **aigp** policy statement on Device PE2.

   ```
   [edit policy-options policy-statement aigp]
   user@PE2# delete term 10
   user@PE2# delete term 20
   ```

3. Configure a recursive route lookup for the route to 66.0.0.0.

   The policy shows the AIGP metric for prefix 66.0.0.0/24 (none) and its recursive next hop. Prefix 66.0.0.0/24 is resolved by 55.0.0.1. Prefix 66.0.0.0/24 does not have its own AIGP metric being originated, but its recursive next hop, 55.0.0.1, has an AIGP value.

   ```
   [edit policy-options policy-statement aigp]
   user@PE2# set term 10 from route-filter 55.0.0.1/24 exact
   user@PE2# set term 10 then aigp-originate distance 20
   user@PE2# set term 10 then next-hop 10.100.1.5
   user@PE2# set term 10 then accept
   user@PE2# set term 20 from route-filter 66.0.0.0/24 exact
   user@PE2# set term 20 then next-hop 55.0.0.1
   ```

```
user@PE2# set term 20 then accept
```

4. On Device PE4, run the **show route 55.0.0.0 extensive** command.

   The value of Metric2 is the IGP metric to the BGP next hop. When Device PE4 readvertises these routes to its IBGP peer, Device PE1, the AIGP metric is the sum of AIGP + its Resolving AIGP metric + Metric2.

   Prefix 55.0.0.0 shows its own IGP metric 20, as defined and advertised by Device PE2. It does not show a resolving AIGP value because it does not have a recursive BGP next hop. The value of Metric2 is 2.

```
user@PE4> show route 55.0.0.0 extensive
inet.0: 31 destinations, 41 routes (31 active, 0 holddown, 0 hidden)
55.0.0.0/24 (1 entry, 1 announced)
TSI:
KRT in-kernel 55.0.0.0/24 -> {indirect(262151)}
Page 0 idx 0 Type 1 val 928d1b8
    Flags: Nexthop Change
    Nexthop: 10.100.1.4
    Localpref: 100
    AS path: [13979] 7018 I
    Communities:
    AIGP: 22
Path 55.0.0.0 from 10.9.9.5 Vector len 4.  Val: 0
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0x925da38
                Next-hop reference count: 4
                Source: 10.9.9.5
                Next hop type: Router, Next hop index: 1004
                Next hop: 10.0.0.22 via fe-1/2/2.10, selected
                Label operation: Push 299888
                Label TTL action: prop-ttl
                Protocol next hop: 10.100.1.5
                Push 299888
                Indirect next hop: 93514d8 262151
                State: <Active Ext>
                Local AS: 13979 Peer AS:  7018
                Age: 22:03:26   Metric2: 2
                AIGP: 20
                Task: BGP_7018.10.9.9.5+58560
                Announcement bits (3): 3-KRT 4-BGP_RT_Background 5-Resolve tree 1
                AS path: 7018 I
                Accepted
                Route Label: 299888
                Localpref: 100
                Router ID: 10.9.9.5
                Indirect next hops: 1
                        Protocol next hop: 10.100.1.5 Metric: 2
                        Push 299888
                        Indirect next hop: 93514d8 262151
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.0.0.22 via fe-1/2/2.10
                        10.100.1.5/32 Originating RIB: inet.0
                          Metric: 2                     Node path count: 1
                          Forwarding nexthops: 1
                                Nexthop: 10.0.0.22 via fe-1/2/2.10
```

5. On Device PE4, run the **show route 66.0.0.0 extensive** command.

Prefix 66.0.0.0/24 shows the Resolving AIGP, which is the sum of its own AIGP metric and its recursive BGP next hop:

66.0.0.1 = 0, 55.0.0.1 = 20, 0+20 = 20

```
user@PE4> show route 66.0.0.0 extensive
inet.0: 31 destinations, 41 routes (31 active, 0 holddown, 0 hidden)
66.0.0.0/24 (1 entry, 1 announced)
TSI:
KRT in-kernel 66.0.0.0/24 -> {indirect(262162)}
Page 0 idx 0 Type 1 val 928cefc
    Flags: Nexthop Change
    Nexthop: 10.100.1.4
    Localpref: 100
    AS path: [13979] 7018 I
    Communities:
Path 66.0.0.0 from 10.9.9.5 Vector len 4.  Val: 0
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0x925d4e0
                Next-hop reference count: 4
                Source: 10.9.9.5
                Next hop type: Router, Next hop index: 1006
                Next hop: 10.0.0.22 via fe-1/2/2.10, selected
                Label operation: Push 299888, Push 299888(top)
                Label TTL action: prop-ttl, prop-ttl(top)
                Protocol next hop: 55.0.0.1
                Push 299888
                Indirect next hop: 9353e88 262162
                State: <Active Ext>
                Local AS: 13979 Peer AS:  7018
                Age: 31:42      Metric2: 2
                Resolving-AIGP: 20
                Task: BGP_7018.10.9.9.5+58560
                Announcement bits (3): 3-KRT 4-BGP_RT_Background 5-Resolve tree 1
                AS path: 7018 I
                Accepted
                Route Label: 299888
                Localpref: 100
                Router ID: 10.9.9.5
                Indirect next hops: 1
                        Protocol next hop: 55.0.0.1 Metric: 2 AIGP: 20
                        Push 299888
                        Indirect next hop: 9353e88 262162
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.0.0.22 via fe-1/2/2.10
                        55.0.0.0/24 Originating RIB: inet.0
                          Metric: 2                     Node path count: 1
                          Indirect nexthops: 1
                                Protocol Nexthop: 10.100.1.5 Metric: 2 Push 299888
                                Indirect nexthop: 93514d8 262151
                                Indirect path forwarding nexthops: 1
                                        Nexthop: 10.0.0.22 via fe-1/2/2.10
                                10.100.1.5/32 Originating RIB: inet.0
                                  Metric: 2                             Node path count: 1
                                  Forwarding nexthops: 1
                                        Nexthop: 10.0.0.22 via fe-1/2/2.10
```

Verifying the Presence of AIGP Attributes in BGP Updates

Purpose If the AIGP attribute is not enabled under BGP (or the **group** or **neighbor** hierarchies), the AIGP attribute is silently discarded. Enable **traceoptions** and include the **packets** flag in the **detail** option in the configuration to confirm the presence of the AIGP attribute in transmitted or received BGP updates. This is useful when debugging AIGP issues.

Action 1. Configure Device PE2 and Device PE4 for **traceoptions**.

```
user@host> show protocols bgp
  traceoptions {
    file bgp size 1m files 5;
    flag packets detail;
  }
```

2. Check the **traceoptions** file on Device PE2.

   The following sample shows Device PE2 advertising prefix 99.0.0.0/24 to Device PE4 (10.9.9.4) with an AIGP metric of 20:

```
user@PE2> show log bgp
Mar 22 09:27:18.982150 BGP SEND 10.9.9.5+49652 -> 10.9.9.4+179
Mar 22 09:27:18.982178 BGP SEND message type 2 (Update) length 70
Mar 22 09:27:18.982198 BGP SEND Update PDU length 70
Mar 22 09:27:18.982248 BGP SEND flags 0x40 code Origin(1): IGP
Mar 22 09:27:18.982273 BGP SEND flags 0x40 code ASPath(2) length 6: 7018
Mar 22 09:27:18.982295 BGP SEND flags 0x80 code AIGP(26): AIGP: 20
Mar 22 09:27:18.982316 BGP SEND flags 0x90 code MP_reach(14): AFI/SAFI 1/4
Mar 22 09:27:18.982341 BGP SEND         nhop 10.100.1.5 len 4
Mar 22 09:27:18.982372 BGP SEND     99.0.0.0/24 (label 301664)
Mar 22 09:27:33.665412 bgp_send: sending 19 bytes to abcd::10:255:170:84
(External AS 13979)
```

3. Verify that the route was received on Device PE4 using the **show route receive-protocol** command.

   AIGP is not enabled on Device PE4, so the AIGP attribute is silently discarded for prefix 99.0.0.0/24 and does not appear in the following output:

```
user@PE4> show route receive-protocol bgp 10.9.9.5 extensive | find 55.0.0.0
* 99.0.0.0/24 (2 entries, 1 announced)
     Accepted
     Route Label: 301728
     Nexthop: 10.100.1.5
     AS path: 7018 I
```

4. Check the **traceoptions** file on Device PE4.

   The following output from the **traceoptions** log shows that the 99.0.0.0/24 prefix was received with the AIGP attribute attached:

```
user@PE4> show log bgp
Mar 22 09:41:39.650295 BGP RECV 10.9.9.5+64690 -> 10.9.9.4+179
Mar 22 09:41:39.650331 BGP RECV message type 2 (Update) length 70
Mar 22 09:41:39.650350 BGP RECV Update PDU length 70
Mar 22 09:41:39.650370 BGP RECV flags 0x40 code Origin(1): IGP
Mar 22 09:41:39.650394 BGP RECV flags 0x40 code ASPath(2) length 6: 7018
Mar 22 09:41:39.650415 BGP RECV flags 0x80 code AIGP(26): AIGP: 20
Mar 22 09:41:39.650436 BGP RECV flags 0x90 code MP_reach(14): AFI/SAFI 1/4
Mar 22 09:41:39.650459 BGP RECV         nhop 10.100.1.5 len 4
```

```
Mar 22 09:41:39.650495 BGP RECV      99.0.0.0/24 (label 301728)
Mar 22 09:41:39.650574 bgp_rcv_nlri: 99.0.0.0/24
Mar 22 09:41:39.650607 bgp_rcv_nlri: 99.0.0.0/24 belongs to meshgroup
Mar 22 09:41:39.650629 bgp_rcv_nlri: 99.0.0.0/24 qualified bnp->ribact 0x0
l2afcb 0x0
```

Meaning Performing this verification helps with AIGP troubleshooting and debugging issues. It enables you to verify which devices in your network send and receive AIGP attributes.

Related Documentation

- Understanding BGP Path Selection on page 197
- Examples: Configuring Internal BGP Peering on page 34

CHAPTER 4

BGP Policy Configuration

Example: Configuring BGP Interactions with IGPs

Understanding Routing Policies

Each routing policy is identified by a policy name. The name can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. To include spaces in the name, enclose the entire name in double quotation marks. Each routing policy name must be unique within a configuration.

Once a policy is created and named, it must be applied before it is active. You apply routing policies using the **import** and **export** statements at the **protocols>*protocol-name*** level in the configuration hierarchy.

In the **import** statement, you list the name of the routing policy to be evaluated when routes are imported into the routing table from the routing protocol.

In the **export** statement, you list the name of the routing policy to be evaluated when routes are being exported from the routing table into a dynamic routing protocol. Only active routes are exported from the routing table.

To specify more than one policy and create a policy chain, you list the policies using a space as a separator. If multiple policies are specified, the policies are evaluated in the order in which they are specified. As soon as an accept or reject action is executed, the policy chain evaluation ends.

Example: Injecting OSPF Routes into the BGP Routing Table

This example shows how to create a policy that injects OSPF routes into the BGP routing table.

Requirements

Before you begin:

- Configure network interfaces.
- Configure external peer sessions. See "Example: Configuring External BGP Point-to-Point Peer Sessions" on page 12.
- Configure interior gateway protocol (IGP) sessions between peers.

Overview

In this example, you create a routing policy called **injectpolicy1** and a routing term called **injectterm1**. The policy injects OSPF routes into the BGP routing table.

Configuration

Configuring the Routing Policy

CLI Quick Configuration

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

```
set policy-options policy-statement injectpolicy1 term injectterm1 from protocol ospf
set policy-options policy-statement injectpolicy1 term injectterm1 from area 0.0.0.1
set policy-options policy-statement injectpolicy1 term injectterm1 then accept
set protocols bgp export injectpolicy1
```

Step-by-Step Procedure

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

To inject OSPF routes into a BGP routing table:

1. Create the policy term.

   ```
   [edit policy-options policy-statement injectpolicy1]
   ```

```
user@host# set term injectterm1
```

2. Specify OSPF as a match condition.

```
[edit policy-options policy-statement injectpolicy1 term injectterm1]
user@host# set from protocol ospf
```

3. Specify the routes from an OSPF area as a match condition.

```
[edit policy-options policy-statement injectpolicy1 term injectterm1]
user@host# set from area 0.0.0.1
```

4. Specify that the route is to be accepted if the previous conditions are matched.

```
[edit policy-options policy-statement injectpolicy1 term injectterm1]
user@host# set then accept
```

5. Apply the routing policy to BGP.

```
[edit]
user@host# set protocols bgp export injectpolicy1
```

Results Confirm your configuration by entering the **show policy-options** and **show protocols bgp** commands from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show policy-options
policy-statement injectpolicy1 {
  term injectterm1 {
    from {
      protocol ospf;
      area 0.0.0.1;
    }
    then accept;
  }
}
```

```
user@host# show protocols bgp
export injectpolicy1;
```

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

Configuring Tracing for the Routing Policy

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

```
set policy-options policy-statement injectpolicy1 term injectterm1 then trace
set routing-options traceoptions file ospf-bgp-policy-log
set routing-options traceoptions file size 5m
set routing-options traceoptions file files 5
set routing-options traceoptions flag policy
```


Step-by-Step Procedure

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

1. Include a trace action in the policy.

```
[edit policy-options policy-statement injectpolicy1 term injectterm1]
user@host# then trace
```

2. Configure the tracing file for the output.

```
[edit routing-options traceoptions]
user@host# set file ospf-bgp-policy-log
user@host# set file size 5m
user@host# set file files 5
user@host# set flag policy
```

Results

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

```
user@host# show policy-options
policy-statement injectpolicy1 {
  term injectterm1 {
    then {
      trace;
    }
  }
}

user@host# show routing-options
traceoptions {
  file ospf-bgp-policy-log size 5m files 5;
  flag policy;
}
```

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

Verification

Confirm that the configuration is working properly.

Verifying That the Expected BGP Routes Are Present

Purpose Verify the effect of the export policy.

Action From operational mode, enter the **show route** command.

Troubleshooting

- Using the show log Command to Examine the Actions of the Routing Policy on page 172

Using the show log Command to Examine the Actions of the Routing Policy

Problem The routing table contains unexpected routes, or routes are missing from the routing table.

Solution If you configure policy tracing as shown in this example, you can run the **show log ospf-bgp-policy-log** command to diagnose problems with the routing policy. The **show log ospf-bgp-policy-log** command displays information about the routes that the **injectpolicy1** policy term analyzes and acts upon.

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

Example: Configuring BGP Route Advertisement

- Understanding Route Advertisement on page 173
- Example: Configuring BGP Prefix-Based Outbound Route Filtering on page 177

Understanding Route Advertisement

All routing protocols use the Junos OS routing table to store the routes that they learn and to determine which routes they should advertise in their protocol packets. Routing policy allows you to control which routes the routing protocols store in and retrieve from the routing table. For information about routing policy, see the *Routing Policy Feature Guide for Routing Devices*.

When configuring BGP routing policy, you can perform the following tasks:

- Applying Routing Policy on page 173
- Setting BGP to Advertise Inactive Routes on page 174
- Configuring BGP to Advertise the Best External Route to Internal Peers on page 174
- Configuring How Often BGP Exchanges Routes with the Routing Table on page 176
- Disabling Suppression of Route Advertisements on page 177

Applying Routing Policy

You define routing policy at the **[edit policy-options]** hierarchy level. To apply policies you have defined for BGP, include the **import** and **export** statements within the BGP configuration.

You can apply policies as follows:

- BGP global **import** and **export** statements—Include these statements at the **[edit protocols bgp]** hierarchy level (for routing instances, include these statements at the **[edit routing-instances *routing-instance-name* protocols bgp]** hierarchy level).
- Group **import** and **export** statements—Include these statements at the **[edit protocols bgp group *group-name*]** hierarchy level (for routing instances, include these statements at the **[edit routing-instances *routing-instance-name* protocols bgp group *group-name*]** hierarchy level).
- Peer **import** and **export** statements—Include these statements at the **[edit protocols bgp group *group-name* neighbor *address*]** hierarchy level (for routing instances, include

these statements at the **[edit routing-instances** *routing-instance-name* **protocols bgp group** *group-name* **neighbor** *address***]** hierarchy level).

A peer-level **import** or **export** statement overrides a group **import** or **export** statement. A group-level **import** or **export** statement overrides a global BGP **import** or **export** statement.

To apply policies, see the following sections:

- Applying Policies to Routes Being Imported into the Routing Table from BGP on page 174
- Applying Policies to Routes Being Exported from the Routing Table into BGP on page 174

Applying Policies to Routes Being Imported into the Routing Table from BGP

To apply policy to routes being imported into the routing table from BGP, include the **import** statement, listing the names of one or more policies to be evaluated:

```
import [ policy-names ];
```

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

If you specify more than one policy, they are evaluated in the order specified, from first to last, and the first matching filter is applied to the route. If no match is found, BGP places into the routing table only those routes that were learned from BGP routing devices.

Applying Policies to Routes Being Exported from the Routing Table into BGP

To apply policy to routes being exported from the routing table into BGP, include the **export** statement, listing the names of one or more policies to be evaluated:

```
export [ policy-names ];
```

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

If you specify more than one policy, they are evaluated in the order specified, from first to last, and the first matching filter is applied to the route. If no routes match the filters, the routing table exports into BGP only the routes that it learned from BGP.

Setting BGP to Advertise Inactive Routes

By default, BGP stores the route information it receives from update messages in the Junos OS routing table, and the routing table exports only active routes into BGP, which BGP then advertises to its peers. To have the routing table export to BGP the best route learned by BGP even if Junos OS did not select it to be an active route, include the **advertise-inactive** statement:

```
advertise-inactive;
```

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

Configuring BGP to Advertise the Best External Route to Internal Peers

In general, deployed BGP implementations do not advertise the external route with the highest local preference value to internal peers unless it is the best route. Although this

behavior was required by an earlier version of the BGP version 4 specification, RFC 1771, it was typically not followed in order to minimize the amount of advertised information and to prevent routing loops. However, there are scenarios in which advertising the best external route is beneficial, in particular, situations that can result in IBGP route oscillation.

In Junos OS Release 9.3 and later, you can configure BGP to advertise the best external route into an internal BGP (IBGP) mesh group, a route reflector cluster, or an autonomous system (AS) confederation, even when the best route is an internal route.

NOTE: In order to configure the advertise-external statement on a route reflector, you must disable intracluster reflection with the no-client-reflect statement.

When a routing device is configured as a route reflector for a cluster, a route advertised by the route reflector is considered internal if it is received from an internal peer with the same cluster identifier or if both peers have no cluster identifier configured. A route received from an internal peer that belongs to another cluster, that is, with a different cluster identifier, is considered external.

In a confederation, when advertising a route to a confederation border router, any route from a different confederation sub-AS is considered external.

You can also configure BGP to advertise the external route only if the route selection process reaches the point where the multiple exit discriminator (MED) metric is evaluated. As a result, an external route with an AS path worse (that is, longer) than that of the active path is not advertised.

Junos OS also provides support for configuring a BGP export policy that matches on the state of an advertised route. You can match on either active or inactive routes. For more information, see the *Routing Policy Feature Guide for Routing Devices*.

To configure BGP to advertise the best external path to internal peers, include the **advertise-external** statement:

```
advertise-external;
```

NOTE: The advertise-external statement is supported at both the group and neighbor level. If you configure the statement at the neighbor level, you must configure it for all neighbors in a group. Otherwise, the group is automatically split into different groups.

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

To configure BGP to advertise the best external path only if the route selection process reaches the point where the MED value is evaluated, include the **conditional** statement:

```
advertise-external {
  conditional;
```

}

Configuring How Often BGP Exchanges Routes with the Routing Table

BGP stores the route information it receives from update messages in the routing table, and the routing table exports active routes from the routing table into BGP. BGP then advertises the exported routes to its peers. By default, the exchange of route information between BGP and the routing table occurs immediately after the routes are received. This immediate exchange of route information might cause instabilities in the network reachability information. To guard against this, you can delay the time between when BGP and the routing table exchange route information.

To configure how often BGP and the routing table exchange route information, include the **out-delay** statement:

```
out-delay seconds;
```

By default, the routing table retains some of the route information learned from BGP. To have the routing table retain all or none of this information, include the **keep** statement:

```
keep (all | none);
```

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

The routing table can retain the route information learned from BGP in one of the following ways:

- Default (omit the **keep** statement)—Keep all route information that was learned from BGP, except for routes whose AS path is looped and whose loop includes the local AS.
- **keep all**—Keep all route information that was learned from BGP.
- **keep none**—Discard routes that were received from a peer and that were rejected by import policy or other sanity checking, such as AS path or next hop. When you configure **keep none** for the BGP session and the inbound policy changes, Junos OS forces readvertisement of the full set of routes advertised by the peer.

In an AS path healing situation, routes with looped paths theoretically could become usable during a soft reconfiguration when the AS path loop limit is changed. However, there is a significant memory usage difference between the default and **keep all**.

Consider the following scenarios:

- A peer readvertises routes back to the peer from which it learned them.

  This can happen in the following cases:

  - Another vendor's routing device advertises the routes back to the sending peer.
  - The Junos OS peer's default behavior of not readvertising routes back to the sending peer is overridden by configuring **advertise-peer-as**.

- A provider edge (PE) routing device discards any VPN route that does not have any of the expected route targets.

When **keep all** is configured, the behavior of discarding routes received in the above scenarios is overridden.

Disabling Suppression of Route Advertisements

Junos OS does not advertise the routes learned from one EBGP peer back to the same external BGP (EBGP) peer. In addition, the software does not advertise those routes back to any EBGP peers that are in the same AS as the originating peer, regardless of the routing instance. You can modify this behavior by including the **advertise-peer-as** statement in the configuration. To disable the default advertisement suppression, include the **advertise-peer-as** statement:

```
advertise-peer-as;
```

NOTE: **The route suppression default behavior is disabled if the as-override statement is included in the configuration.**

If you include the **advertise-peer-as** statement in the configuration, BGP advertises the route regardless of this check.

To restore the default behavior, include the **no-advertise-peer-as** statement in the configuration:

```
no-advertise-peer-as;
```

If you include both the **as-override** and **no-advertise-peer-as** statements in the configuration, the **no-advertise-peer-as** statement is ignored. You can include these statements at multiple hierarchy levels.

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

Example: Configuring BGP Prefix-Based Outbound Route Filtering

This example shows how to configure a Juniper Networks router to accept route filters from remote peers and perform outbound route filtering using the received filters.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol (IGP).

Overview

You can configure a BGP peer to accept route filters from remote peers and perform outbound route filtering using the received filters. By filtering out unwanted updates, the sending peer saves resources needed to generate and transmit updates, and the receiving peer saves resources needed to process updates. This feature can be useful, for example, in a virtual private network (VPN) in which subsets of customer edge (CE) devices are not capable of processing all the routes in the VPN. The CE devices can use prefix-based outbound route filtering to communicate to the provider edge (PE) routing device to transmit only a subset of routes, such as routes to the main data centers only.

The maximum number of prefix-based outbound route filters that a BGP peer can accept is 5000. If a remote peer sends more than 5000 outbound route filters to a peer address, the additional filters are discarded, and a system log message is generated.

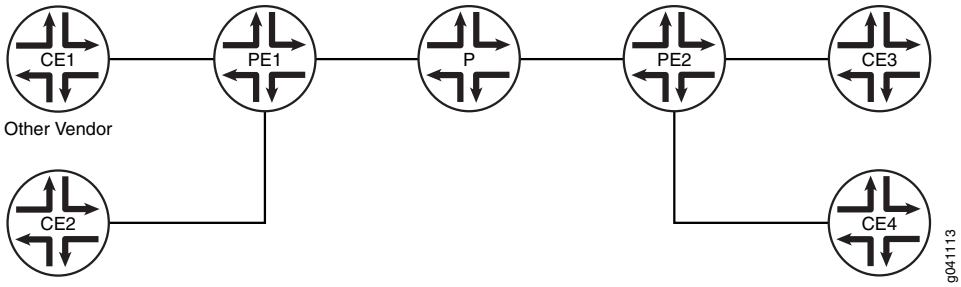
You can configure interoperability for the routing device as a whole or for specific BGP groups or peers only.

Topology

In the sample network, Device CE1 is a router from another vendor. The configuration shown in this example is on Juniper Networks Router PE1.

Figure 17 on page 178 shows the sample network.

Figure 17: BGP Prefix-Based Outbound Route Filtering



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.

PE1

```
set protocols bgp group cisco-peers type external
set protocols bgp group cisco-peers description “to CE1”
set protocols bgp group cisco-peers local-address 192.168.165.58
set protocols bgp group cisco-peers peer-as 35
set protocols bgp group cisco-peers outbound-route-filter bgp-orf-cisco-mode
set protocols bgp group cisco-peers outbound-route-filter prefix-based accept inet
set protocols bgp group cisco-peers neighbor 192.168.165.56
set routing-options autonomous-system 65500
```

Step-by-Step Procedure

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

To configure Router PE1 to accept route filters from Device CE1 and perform outbound route filtering using the received filters:

1. Configure the local autonomous system.

```
[edit routing-options]
user@PE1# set autonomous-system 65500
```

2. Configure external peering with Device CE1.

```
[edit protocols bgp group cisco-peers]
user@PE1# set type external
user@PE1# set description “to CE1”
user@PE1# set local-address 192.168.165.58
user@PE1# set peer-as 35
user@PE1# set neighbor 192.168.165.56
```

3. Configure Router PE1 to accept IPv4 route filters from Device CE1 and perform outbound route filtering using the received filters.

```
[edit protocols bgp group cisco-peers]
user@PE1# set outbound-route-filter prefix-based accept inet
```

4. (Optional) Enable interoperability with routing devices that use the vendor-specific compatibility code of 130 for outbound route filters and the code type of 128.

   The IANA standard code is 3, and the standard code type is 64.

```
[edit protocols bgp group cisco-peers]
user@PE1# set outbound-route-filter bgp-orf-cisco-mode
```

Results

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

```
user@PE1# show protocols
group cisco-peers {
   type external;
   description “to CE1”;
   local-address 192.168.165.58;
   peer-as 35;
   outbound-route-filter {
      bgp-orf-cisco-mode;
      prefix-based {
         accept {
            inet;
         }
      }
   }
   neighbor 192.168.165.56;
}

user@PE1# show routing-options
autonomous-system 65500;
```

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

Verification

Confirm that the configuration is working properly.

- Verifying the Outbound Route Filter on page 180
- Verifying the BGP Neighbor Mode on page 180

Verifying the Outbound Route Filter

Purpose Display information about the prefix-based outbound route filter received from Device CE1.

Action From operational mode, enter the **show bgp neighbor orf detail** command.

```
user@PE1> show bgp neighbor orf 192.168.165.56 detail
   Peer: 192.168.165.56 Type: External
   Group: cisco-peers

   inet-unicast
    Filter updates recv:          4 Immediate:          0
    Filter: prefix-based            receive
            Updates recv:          4
      Received filter entries:
        seq 10 2.2.0.0/16 deny minlen 0 maxlen 0
        seq 20 3.3.0.0/16 deny minlen 24 maxlen 0
        seq 30 4.4.0.0/16 deny minlen 0 maxlen 28
        seq 40 5.5.0.0/16 deny minlen 24 maxlen 28
```

Verifying the BGP Neighbor Mode

Purpose Verify that the **bgp-orf-cisco-mode** setting is enabled for the peer by making sure that the **ORFCiscoMode** option is displayed in the **show bgp neighbor** command output.

Action From operational mode, enter the **show bgp neighbor** command.

```
user@PE1> show bgp neighbor
Peer: 192.168.165.56 AS 35       Local: 192.168.165.58 AS 65500
  Type: External    State: Active         Flags: <>
  Last State: Idle          Last Event: Start
  Last Error: None
  Export: [ adv_stat ]
  Options: <Preference LocalAddress AddressFamily PeerAS Refresh>
  Options: <ORF ORFCiscoMode>
  Address families configured: inet-unicast
  Local Address: 192.168.165.58 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Trace options: detail open detail refresh
  Trace file: /var/log/orf size 5242880 files 20
```

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*
- *Example: Configuring BGP to Advertise the Best External Route to Internal Peers*
- *Example: Setting BGP to Advertise Inactive Routes*

Example: Configuring EBGP Multihop

Understanding BGP Multihop

When external BGP (EBGP) peers are not directly connected to each other, they must cross one or more non-BGP routers to reach each other. Configuring multihop EBGP enables the peers to pass through the other routers to form peer relationships and exchange update messages. This type of configuration is typically used when a Juniper Networks routing device needs to run EBGP with a third-party router that does not allow direct connection of the two EBGP peers. EBGP multihop enables a neighbor connection between two EBGP peers that do not have a direct connection.

Example: Configuring EBGP Multihop Sessions

This example shows how to configure an external BGP (EBGP) peer that is more than one hop away from the local router. This type of session is called a *multihop* BGP session.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

The configuration to enable multihop EBGP sessions requires connectivity between the two EBGP peers. This example uses static routes to provide connectivity between the devices.

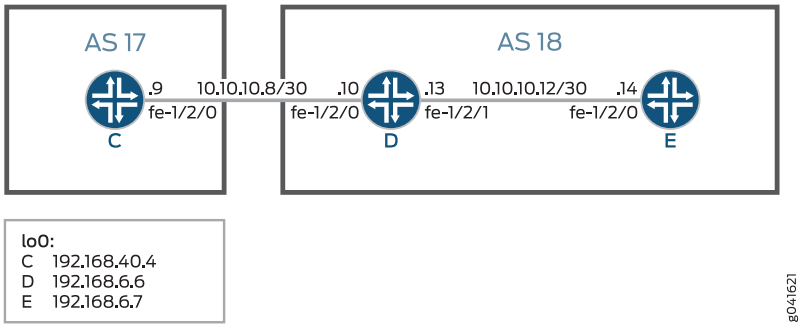
Unlike directly connected EBGP sessions in which physical address are typically used in the **neighbor** statements, you must use loopback interface addresses for multihop EBGP by specifying the loopback interface address of the indirectly connected peer. In this way, EBGP multihop is similar to internal BGP (IBGP).

Finally, you must add the **multihop** statement. Optionally, you can set a maximum time-to-live (TTL) value with the **ttl** statement. The TTL is carried in the IP header of BGP packets. If you do not specify a TTL value, the system's default maximum TTL value is used. The default TTL value is 64 for multihop EBGP sessions. Another option is to retain the BGP next-hop value for route advertisements by including the **no-nexthop-change** statement.

Figure 18 on page 182 shows a typical EBGP multihop network.

Device C and Device E have an established EBGP session. Device D is not a BGP-enabled device. All of the devices have connectivity via static routes.

Figure 18: Typical Network with EBGP Multihop Sessions



Configuration

CLI Quick Configuration

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

Device C

```
set interfaces fe-1/2/0 unit 9 description to-D
set interfaces fe-1/2/0 unit 9 family inet address 10.10.10.9/30
set interfaces lo0 unit 3 family inet address 192.168.40.4/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers multihop ttl 2
set protocols bgp group external-peers local-address 192.168.40.4
set protocols bgp group external-peers export send-static
set protocols bgp group external-peers peer-as 18
set protocols bgp group external-peers neighbor 192.168.6.7
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 10.10.10.14/32 next-hop 10.10.10.10
set routing-options static route 192.168.6.7/32 next-hop 10.10.10.10
set routing-options router-id 192.168.40.4
set routing-options autonomous-system 17
```

Device D

```
set interfaces fe-1/2/0 unit 10 description to-C
set interfaces fe-1/2/0 unit 10 family inet address 10.10.10.10/30
set interfaces fe-1/2/1 unit 13 description to-E
set interfaces fe-1/2/1 unit 13 family inet address 10.10.10.13/30
set interfaces lo0 unit 4 family inet address 192.168.6.6/32
set routing-options static route 192.168.40.4/32 next-hop 10.10.10.9
set routing-options static route 192.168.6.7/32 next-hop 10.10.10.14
set routing-options router-id 192.168.6.6
```

Device E

```
set interfaces fe-1/2/0 unit 14 description to-D
set interfaces fe-1/2/0 unit 14 family inet address 10.10.10.14/30
set interfaces lo0 unit 5 family inet address 192.168.6.7/32
set protocols bgp group external-peers multihop ttl 2
set protocols bgp group external-peers local-address 192.168.6.7
set protocols bgp group external-peers export send-static
set protocols bgp group external-peers peer-as 17
```

```
set protocols bgp group external-peers neighbor 192.168.40.4
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 10.10.10.8/30 next-hop 10.10.10.13
set routing-options static route 192.168.40.4/32 next-hop 10.10.10.13
set routing-options router-id 192.168.6.7
set routing-options autonomous-system 18
```

Device C

Step-by-Step Procedure

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

To configure Device C:

1. Configure the interface to the directly connected device (to-D), and configure the loopback interface.

   ```
   [edit interfaces fe-1/2/0 unit 9]
   user@C# set description to-D
   user@C# set family inet address 10.10.10.9/30

   [edit interfaces lo0 unit 3]
   user@C# set family inet address 192.168.40.4/32
   ```

2. Configure an EBGP session with Device E.

   The **neighbor** statement points to the loopback interface on Device E.

   ```
   [edit protocols bgp group external-peers]
   user@C# set type external
   user@C# set local-address 192.168.40.4
   user@C# set export send-static
   user@C# set peer-as 18
   user@C# set neighbor 192.168.6.7
   ```

3. Configure the multihop statement to enable Device C and Device E to become EBGP peers.

   Because the peers are two hops away from each other, the example uses the **ttl 2** statement.

   ```
   [edit protocols bgp group external-peers]
   user@C# set multihop ttl 2
   ```

4. Configure connectivity to Device E, using static routes.

   You must configure a route to both the loopback interface address and to the address on the physical interface.

   ```
   [edit routing-options]
   user@C# set static route 10.10.10.14/32 next-hop 10.10.10.10
   user@C# set static route 192.168.6.7/32 next-hop 10.10.10.10
   ```

5. Configure the local router ID and the autonomous system (AS) number.

   ```
   [edit routing-options]
   user@C# set router-id 192.168.40.4
   ```

```
user@C# set autonomous-system 17
```

6. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-static term 1]
user@C# set from protocol static
user@C# set then accept
```

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

```
user@C# show interfaces
fe-1/2/0 {
    unit 9 {
        description to-D;
        family inet {
            address 10.10.10.9/30;
        }
    }
}
lo0 {
    unit 3 {
        family inet {
            address 192.168.40.4/32;
        }
    }
}
```

```
user@C# show protocols
bgp {
    group external-peers {
        type external;
        multihop {
            ttl 2;
        }
        local-address 192.168.40.4;
        export send-static;
        peer-as 18;
        neighbor 192.168.6.7;
    }
}
```

```
user@C# show policy-options
policy-statement send-static {
    term 1 {
        from protocol static;
        then accept;
    }
}
```

```
user@C# show routing-options
static {
```

```
    route 10.10.10.14/32 next-hop 10.10.10.10;
    route 192.168.6.7/32 next-hop 10.10.10.10;
  }
  router-id 192.168.40.4;
  autonomous-system 17;
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat these steps for all BFD sessions in the topology.

Configuring Device D

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 Device D:

1. Set the CLI to Device D.

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

2. Configure the interfaces to the directly connected devices, and configure a loopback interface.

   ```
   [edit interfaces fe-1/2/0 unit 10]
   user@D# set description to-C
   user@D# set family inet address 10.10.10.10/30

   [edit interfaces fe-1/2/1 unit 13]
   user@D# set description to-E
   user@D# set family inet address 10.10.10.13/30

   [edit interfaces lo0 unit 4]
   user@D# set family inet address 192.168.6.6/32
   ```

3. Configure connectivity to the other devices using static routes to the loopback interface addresses.

   On Device D, you do not need static routes to the physical addresses because Device D is directly connected to Device C and Device E.

   ```
   [edit routing-options]
   user@D# set static route 192.168.40.4/32 next-hop 10.10.10.9
   user@D# set static route 192.168.6.7/32 next-hop 10.10.10.14
   ```

4. Configure the local router ID.

   ```
   [edit routing-options]
   user@D# set router-id 192.168.6.6
   ```

Results

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

```
user@D# show interfaces
fe-1/2/0 {
  unit 10 {
```

```
        description to-C;
        family inet {
          address 10.10.10.10/30;
        }
      }
    }
    fe-1/2/1 {
      unit 13 {
        description to-E;
        family inet {
          address 10.10.10.13/30;
        }
      }
    }
    lo0 {
      unit 4 {
        family inet {
          address 192.168.6.6/32;
        }
      }
    }
```

```
user@D# show protocols
```

```
user@D# show routing-options
static {
  route 192.168.40.4/32 next-hop 10.10.10.9;
  route 192.168.6.7/32 next-hop 10.10.10.14;
}
router-id 192.168.6.6;
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat these steps for all BFD sessions in the topology.

Configuring Device E

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 Device E:

1. Set the CLI to Device E.

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

2. Configure the interface to the directly connected device (to-D), and configure the loopback interface.

   ```
   [edit interfaces fe-1/2/0 unit 14]
   user@E# set description to-D
   user@E# set family inet address 10.10.10.14/30

   [edit interfaces lo0 unit 5]
   user@E# set family inet address 192.168.6.7/32
   ```

3. Configure an EBGP session with Device E.

The **neighbor** statement points to the loopback interface on Device C.

```
[edit protocols bgp group external-peers]
user@E# set local-address 192.168.6.7
user@E# set export send-static
user@E# set peer-as 17
user@E# set neighbor 192.168.40.4
```

4. Configure the **multihop** statement to enable Device C and Device E to become EBGP peers.

   Because the peers are two hops away from each other, the example uses the **ttl 2** statement.

   ```
   [edit protocols bgp group external-peers]
   user@E# set multihop ttl 2
   ```

5. Configure connectivity to Device E, using static routes.

   You must configure a route to both the loopback interface address and to the address on the physical interface.

   ```
   [edit routing-options]
   user@E# set static route 10.10.10.8/30 next-hop 10.10.10.13
   user@E# set static route 192.168.40.4/32 next-hop 10.10.10.13
   ```

6. Configure the local router ID and the autonomous system (AS) number.

   ```
   [edit routing-options]
   user@E# set router-id 192.168.6.7
   user@E# set autonomous-system 18
   ```

7. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

   ```
   [edit policy-options policy-statement send-static term 1]
   user@E# set from protocol static
   user@E# set then accept
   ```

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

```
user@E# show interfaces
fe-1/2/0 {
    unit 14 {
        description to-D;
        family inet {
            address 10.10.10.14/30;
        }
    }
}
lo0 {
    unit 5 {
        family inet {
            address 192.168.6.7/32;
```

```
    }
  }
}
```

```
user@E# show protocols
bgp {
  group external-peers {
    multihop {
      ttl 2;
    }
    local-address 192.168.6.7;
    export send-static;
    peer-as 17;
    neighbor 192.168.40.4;
  }
}
```

```
user@E# show policy-options
policy-statement send-static {
  term 1 {
    from protocol static;
    then accept;
  }
}
```

```
user@E# show routing-options
static {
  route 10.10.10.8/30 next-hop 10.10.10.13;
  route 192.168.40.4/32 next-hop 10.10.10.13;
}
router-id 192.168.6.7;
autonomous-system 18;
```

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

Verification

Confirm that the configuration is working properly.

- Verifying Connectivity on page 188
- Verifying That BGP Sessions Are Established on page 189
- Viewing Advertised Routes on page 189

Verifying Connectivity

Purpose Make sure that Device C can ping Device E, specifying the loopback interface address as the source of the ping request.

The loopback interface address is the source address that BGP will use.

Action From operational mode, enter the **ping 10.10.10.14 source 192.168.40.4** command from Device C, and enter the **ping 10.10.10.9 source 192.168.6.7** command from Device E.

```
user@C> ping 10.10.10.14 source 192.168.40.4

PING 10.10.10.14 (10.10.10.14): 56 data bytes
```

```
64 bytes from 10.10.10.14: icmp_seq=0 ttl=63 time=1.262 ms
64 bytes from 10.10.10.14: icmp_seq=1 ttl=63 time=1.202 ms
^C
--- 10.10.10.14 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.202/1.232/1.262/0.030 ms

user@E> ping 10.10.10.9 source 192.168.6.7

PING 10.10.10.9 (10.10.10.9): 56 data bytes
64 bytes from 10.10.10.9: icmp_seq=0 ttl=63 time=1.255 ms
64 bytes from 10.10.10.9: icmp_seq=1 ttl=63 time=1.158 ms
^C
--- 10.10.10.9 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.158/1.206/1.255/0.049 ms
```

Meaning The static routes are working if the pings work.

Verifying That BGP Sessions Are Established

Purpose Verify that the BGP sessions are up.

Action From operational mode, enter the show bgp summary command.

```
user@C> show bgp summary

Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 2          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.168.6.7              18        147        147       0       1     1:04:27
0/2/2/0              0/0/0/0

user@E> show bgp summary

Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 2          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.168.40.4             17        202        202       0       1     1:02:18
0/2/2/0              0/0/0/0
```

Meaning The output shows that both devices have one peer each. No peers are down.

Viewing Advertised Routes

Purpose Check to make sure that routes are being advertised by BGP.

Action From operational mode, enter the **show route advertising-protocol bgp *neighbor*** command.

```
user@C> show route advertising-protocol bgp 192.168.6.7

inet.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
```

```
* 10.10.10.14/32          Self                                  I
* 192.168.6.7/32          Self                                  I

user@E> show route advertising-protocol bgp 192.168.40.4

inet.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.10.10.8/30           Self                                    I
* 192.168.40.4/32         Self                                    I
```

Meaning The **send-static** routing policy is exporting the static routes from the routing table into BGP. BGP is advertising these routes between the peers because the BGP peer session is established.

Related Documentation

- Examples: Configuring External BGP Peering on page 11
- *BGP Configuration Overview*

Example: Configuring BGP Route Preference (Administrative Distance)

- Understanding Route Preference Values on page 190
- Example: Configuring the Preference Value for BGP Routes on page 191

Understanding Route Preference Values

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 4 on page 190 lists the default preference values.

Table 4: 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*
RSVP-signaled LSPs	7	RSVP **preference** as described in the *Junos OS MPLS Applications Library for Routing Devices*
LDP-signaled LSPs	9	LDP **preference**, as described in the *Junos OS MPLS Applications Library for Routing Devices*
OSPF internal route	10	OSPF *preference*
IS-IS Level 1 internal route	15	IS-IS *preference*

Table 4: Default Route Preference Values *(continued)*

How Route Is Learned	Default Preference	Statement to Modify Default Preference
IS-IS Level 2 internal route	18	IS-IS *preference*
Redirects	30	–
Kernel	40	–
SNMP	50	–
Router discovery	55	–
RIP	100	RIP *preference*
RIPng	100	RIPng *preference*
PIM	105	*Multicast Protocols Feature Guide for Routing Devices*
DVMRP	110	*Multicast Protocols Feature Guide for Routing Devices*
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 for Routing Devices*

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.

Example: Configuring the Preference Value for BGP Routes

This example shows how to specify the preference for routes learned from BGP. Routing information can be learned from multiple sources. To break ties among equally specific routes learned from multiple sources, each source has a preference value. Routes that are learned through explicit administrative action, such as static routes, are preferred

over routes learned from a routing protocol, such as BGP or OSPF. This concept is called *administrative distance* by some vendors.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

Routing information can be learned from multiple sources, such as through static configuration, BGP, or an interior gateway protocol (IGP). When Junos OS determines a route's preference to become the active route, it selects the route with the lowest preference as the active route and installs this route into the forwarding table. By default, the routing software assigns a preference of 170 to routes that originated from BGP. Of all the routing protocols, BGP has the highest default preference value, which means that routes learned by BGP are the least likely to become the active route.

Some vendors have a preference (distance) of 20 for external BGP (EBGP) and a distance of 200 for internal BGP (IGBP). Junos OS uses the same value (170) for both EBGP and IBGP. However, this difference between vendors has no operational impact because Junos OS always prefers EBGP routes over IBGP routes.

Another area in which vendors differ is in regard to IGP distance compared to BGP distance. For example, some vendors assign a distance of 110 to OSPF routes. This is higher than the EBGP distance of 20 , and results in the selection of an EBGP route over an equivalent OSPF route. In the same scenario, Junos OS chooses the OSPF route, because of the default preference 10 for an internal OSPF route and 150 for an external OSPF route, which are both lower than the 170 preference assigned to all BGP routes.

In a multivendor environment, you might want to change the preference value for BGP routes so that Junos OS chooses an EBGP route instead of an OSPF route. To accomplish this goal, one option is to include the preference statement in the EBGP configuration. To modify the default BGP preference value, include the **preferece** statement, specifying a value from 0 through 4,294,967,295 ($2^{32} - 1$).

TIP: Another way to achieve multivendor compatibility is to include the advertise-inactive statement in the EBGP configuration. This causes the routing table to export to BGP the best route learned by BGP even if Junos OS did not select it to be an active route. By default, BGP stores the route information it receives from update messages in the Junos OS routing table, and the routing table exports only active routes into BGP, which BGP then advertises to its peers. The advertise-inactive statement causes Junos OS to advertise the best BGP route that is inactive because of IGP preference. When

you use the advertise-inactive statement, the Junos OS device uses the OSPF route for forwarding, and the other vendor's device uses the EBGP route for forwarding. However, from the perspective of an EBGP peer in a neighboring AS, both vendors' devices appear to behave the same way.

Topology

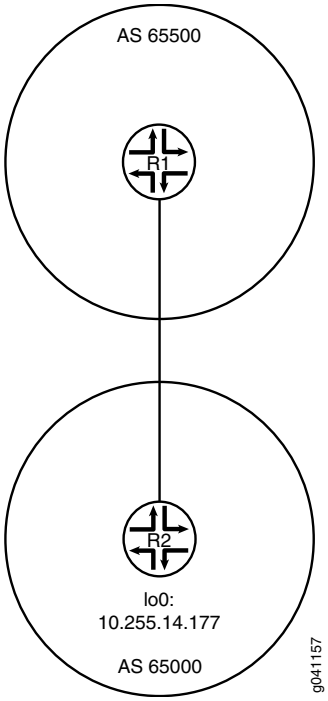
In the sample network, Device R1 and Device R2 have EBGP routes to each other and also OSPF routes to each other.

This example shows the routing tables in the following cases:

- Accept the default preference values of 170 for BGP and 10 for OSPF.
- Change the BGP preference to 8.

Figure 19 on page 193 shows the sample network.

Figure 19: BGP Preference Value Topology



Configuration

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 4 family inet address 1.12.0.1/30
set interfaces lo0 unit 2 family inet address 10.255.71.24/32
set protocols bgp export send-direct
```

```
set protocols bgp group ext type external
set protocols bgp group ext preference 8
set protocols bgp group ext peer-as 65000
set protocols bgp group ext neighbor 1.12.0.2
set protocols ospf area 0.0.0.0 interface fe-1/2/0.4
set protocols ospf area 0.0.0.0 interface 10.255.71.24
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 65500
```

Device R2

```
set interfaces fe-1/2/0 unit 6 family inet address 1.12.0.2/30
set interfaces lo0 unit 3 family inet address 10.255.14.177/32
set protocols bgp export send-direct
set protocols bgp group ext type external
set protocols bgp group ext peer-as 65500
set protocols bgp group ext neighbor 1.12.0.1
set protocols ospf area 0.0.0.0 interface fe-1/2/0.6
set protocols ospf area 0.0.0.0 interface 10.255.14.177
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set routing-options autonomous-system 65000
```

Step-by-Step Procedure

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

To configure Device R1:

1. Configure the interfaces.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 4 family inet address 1.12.0.1/30
user@R1# set lo0 unit 2 family inet address 10.255.71.24/32
```

2. Configure the local autonomous system.

```
[edit routing-options]
user@R1# set autonomous-system 65500
```

3. Configure the external peering with Device R2.

```
[edit protocols bgp]
user@R1# set export send-direct
user@R1# set group ext type external
user@R1# set group ext preference 8
user@R1# set group ext peer-as 65000
user@R1# set group ext neighbor 1.12.0.2
```

4. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface fe-1/2/0.4
user@R1# set interface 10.255.71.24
```

5. Configure the routing policy.

```
[edit policy-options policy-statement send-direct term 1]
user@R1# set from protocol direct
user@R1# set then accept
```

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

```
user@R1# show interfaces
fe-1/2/0 {
    unit 4 {
        family inet {
            address 1.12.0.1/30;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 10.255.71.24/32;
        }
    }
}
```

```
user@R1# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}
```

```
user@R1# show protocols
protocols {
    bgp {
        export send-direct;
        group ext {
            type external;
            preference 8;
            peer-as 65000;
            neighbor 1.12.0.2;
        }
    }
    ospf {
        area 0.0.0.0 {
            interface fe-1/2/0.4;
            interface 10.255.71.24;
        }
    }
}
```

```
user@R1# show routing-options
autonomous-system 65500;
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat these steps on Device R2.

Verification

Confirm that the configuration is working properly.

Verifying the Preference

Purpose Make sure that the routing tables on Device R1 and Device R2 reflect the fact that Device R1 is using the configured EBGP preference of 8, and Device R2 is using the default EBGP preference of 170.

Action From operational mode, enter the **show route** command.

```
user@R1> show route
inet.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.12.0.0/30        *[Direct/0] 3d 07:03:01
                    > via fe-1/2/0.4
                    [BGP/8] 01:04:49, localpref 100
                      AS path: 65000 I
                    > to 1.12.0.2 via fe-1/2/0.4
1.12.0.1/32        *[Local/0] 3d 07:03:01
                      Local via fe-1/2/0.4
10.255.14.177/32   *[BGP/8] 01:04:49, localpref 100
                      AS path: 65000 I
                    > to 1.12.0.2 via fe-1/2/0.4
                    [OSPF/10] 3d 07:02:16, metric 1
                    > to 1.12.0.2 via fe-1/2/0.4
10.255.71.24/32    *[Direct/0] 3d 07:03:01
                    > via lo0.2
224.0.0.5/32       *[OSPF/10] 5d 03:42:16, metric 1
                      MultiRecv

user@R2> show route
inet.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.12.0.0/30        *[Direct/0] 3d 07:03:30
                    > via fe-1/2/0.6
                    [BGP/170] 00:45:36, localpref 100
                      AS path: 65500 I
                    > to 1.12.0.1 via fe-1/2/0.6
1.12.0.2/32        *[Local/0] 3d 07:03:30
                      Local via fe-1/2/0.6
10.255.14.177/32   *[Direct/0] 3d 07:03:30
                    > via lo0.3
10.255.71.24/32    *[OSPF/10] 3d 07:02:45, metric 1
                    > to 1.12.0.1 via fe-1/2/0.6
                    [BGP/170] 00:45:36, localpref 100
                      AS path: 65500 I
                    > to 1.12.0.1 via fe-1/2/0.6
224.0.0.5/32       *[OSPF/10] 5d 03:42:45, metric 1
                      MultiRecv
```

Meaning The output shows that on Device R1, the active path to Device R2's loopback interface (10.255.14.177/32) is a BGP route. The output also shows that on Device R2, the active path to Device R1's loopback interface (10.255.71.24/32) is an OSPF route.

Related Documentation

- *Route Preferences Overview*
- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

Example: Configuring BGP Path Selection

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.

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: **The as-path-ignore option is not supported for routing instances.**

To configure routing table path selection behavior, include the **path-selection** statement:

```
path-selection {
  (always-compare-med | cisco-non-deterministic | external-router-id);
  as-path-ignore;
  med-plus-igp {
    igp-multiplier number;
    med-multiplier number;
  }
}
```

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

Routing table path selection can be configured in one of the following ways:

- 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.

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.

Example: Ignoring the AS Path Attribute When Selecting the Best Path

If multiple BGP routes to the same destination exist, BGP selects the best path based on the route attributes of the paths. One of the route attributes that affects the best-path decision is the length of the AS paths of each route. Routes with shorter AS paths are preferred over those with longer AS paths. Although not typically practical, some scenarios might require that the AS path length be ignored in the route selection process. This example shows how to configure a routing device to ignore the AS path attribute.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

On externally connected routing devices, the purpose of skipping the AS path comparison might be to force an external BGP (EBGP) versus internal BGP (IBGP) decision to remove traffic from your network as soon as possible. On internally connected routing devices, you might want your IBGP-only routers to default to the local externally connected gateway. The local IBGP-only (internal) routers skip the AS path comparison and move down the decision tree to use the closest interior gateway protocol (IGP) gateway (lowest IGP metric). Doing this might be an effective way to force these routers to use a LAN connection instead of their WAN connection.

CAUTION: **When you include the as-path-ignore statement on a routing device in your network, you might need to include it on all other BGP-enabled devices in your network to prevent routing loops and convergence issues. This is especially true for IBGP path comparisons.**

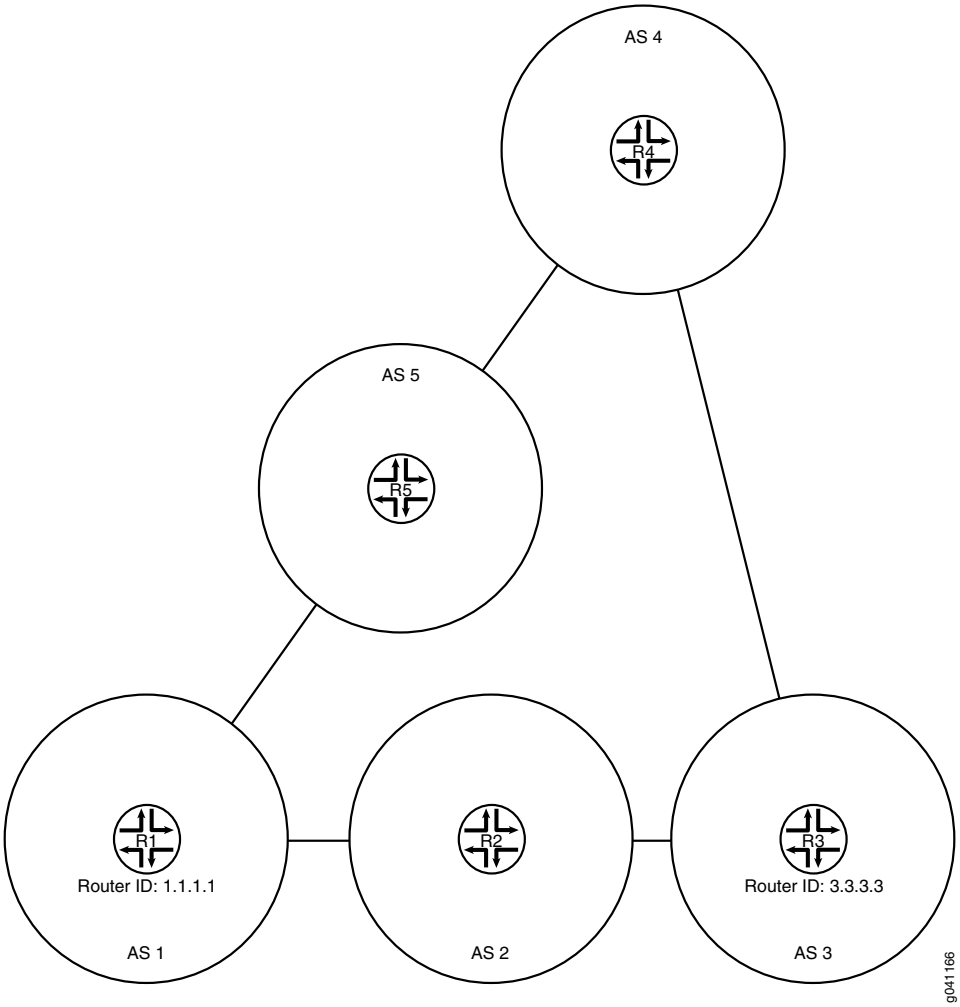
In this example, Device R2 is learning about the loopback interface address on Device R4 (4.4.4.4/32) from Device R1 and Device R3. Device R1 is advertising 4.4.4.4/32 with an AS-path of 1 5 4, and Device R3 is advertising 4.4.4.4/32 with an AS-path of 3 4. Device R2 selects the path for 4.4.4.4/32 from Device R3 as the best path because the AS path is shorter than the AS path from Device R1.

This example modifies the BGP configuration on Device R2 so that the AS-path length is not used in the best-path selection.

Device R1 has a lower router ID (1.1.1.1) than Device R3 (1.1.1.1). If all other path selection criteria are equal (or, as in this case, ignored), the route learned from Device R1 is used. Because the AS-path attribute is being ignored, the best path is toward Device R1 because of its lower router ID value.

Figure 20 on page 202 shows the sample topology.

Figure 20: Topology for Ignoring the AS-Path Lengh



Configuration

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 192.168.10.1/24
set interfaces fe-1/2/1 unit 10 family inet address 192.168.50.2/24
set interfaces lo0 unit 1 family inet address 1.1.1.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.10.2 peer-as 2
set protocols bgp group ext neighbor 192.168.50.1 peer-as 5
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
```

```
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.20.0/24 next-hop 192.168.10.2
set routing-options static route 192.168.30.0/24 next-hop 192.168.10.2
set routing-options static route 192.168.40.0/24 next-hop 192.168.50.1
set routing-options router-id 1.1.1.1
set routing-options autonomous-system 1
```

Device R2

```
set interfaces fe-1/2/0 unit 2 family inet address 192.168.10.2/24
set interfaces fe-1/2/1 unit 3 family inet address 192.168.20.2/24
set interfaces lo0 unit 2 family inet address 2.2.2.2/32
set protocols bgp path-selection as-path-ignore
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.10.1 peer-as 1
set protocols bgp group ext neighbor 192.168.20.1 peer-as 3
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.50.0/24 next-hop 192.168.10.1
set routing-options static route 192.168.40.0/24 next-hop 192.168.10.1
set routing-options static route 192.168.30.0/24 next-hop 192.168.20.1
set routing-options router-id 2.2.2.2
set routing-options autonomous-system 2
```

Device R3

```
set interfaces fe-1/2/0 unit 4 family inet address 192.168.20.1/24
set interfaces fe-1/2/1 unit 5 family inet address 192.168.30.1/24
set interfaces lo0 unit 3 family inet address 1.1.1.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.20.2 peer-as 2
set protocols bgp group ext neighbor 192.168.30.2 peer-as 4
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.20.2
set routing-options static route 192.168.50.0/24 next-hop 192.168.20.2
set routing-options static route 192.168.40.0/24 next-hop 192.168.30.2
set routing-options router-id 3.3.3.3
set routing-options autonomous-system 3
```

Device R4

```
set interfaces fe-1/2/0 unit 6 family inet address 192.168.30.2/24
set interfaces fe-1/2/1 unit 7 family inet address 192.168.40.1/24
set interfaces lo0 unit 4 family inet address 4.4.4.4/32
```

```
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.30.1 peer-as 3
set protocols bgp group ext neighbor 192.168.40.2 peer-as 5
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.40.2
set routing-options static route 192.168.50.0/24 next-hop 192.168.40.2
set routing-options static route 192.168.40.0/24 next-hop 192.168.30.1
set routing-options router-id 4.4.4.4
set routing-options autonomous-system 4
```

Device R5

```
set interfaces fe-1/2/0 unit 8 family inet address 192.168.40.2/24
set interfaces fe-1/2/1 unit 9 family inet address 192.168.50.1/24
set interfaces lo0 unit 5 family inet address 5.5.5.5/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext export send-local
set protocols bgp group ext neighbor 192.168.40.1 peer-as 4
set protocols bgp group ext neighbor 192.168.50.2 peer-as 1
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-local term 1 from protocol local
set policy-options policy-statement send-local term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.50.2
set routing-options static route 192.168.20.0/24 next-hop 192.168.50.2
set routing-options static route 192.168.30.0/24 next-hop 192.168.40.1
set routing-options router-id 5.5.5.5
set routing-options autonomous-system 5
```

Configuring Device R2

Step-by-Step Procedure

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

To configure Device R2:

1. Configure the interfaces.

   ```
   [edit interfaces]
   user@R2# set fe-1/2/0 unit 2 family inet address 192.168.10.2/24
   user@R2# set fe-1/2/1 unit 3 family inet address 192.168.20.2/24
   user@R2# set lo0 unit 2 family inet address 2.2.2.2/32
   ```

2. Configure EBGP.

   ```
   [edit protocols bgp group ext]
   ```

```
user@R2# set type external
user@R2# set export send-direct
user@R2# set export send-static
user@R2# set export send-local
user@R2# set neighbor 192.168.10.1 peer-as 1
user@R2# set neighbor 192.168.20.1 peer-as 3
```

3. Configure the autonomous system (AS) path attribute to be ignored in the Junos OS path selection algorithm.

```
[edit protocols bgp]
user@R2# set path-selection as-path-ignore
```

4. Configure the routing policy.

```
[edit policy-options]
user@R2# set policy-statement send-direct term 1 from protocol direct
user@R2# set policy-statement send-direct term 1 then accept
user@R2# set policy-statement send-local term 1 from protocol local
user@R2# set policy-statement send-local term 1 then accept
user@R2# set policy-statement send-static term 1 from protocol static
user@R2# set policy-statement send-static term 1 then accept
```

5. Configure some static routes.

```
[edit routing-options static]
user@R2# set route 192.168.50.0/24 next-hop 192.168.10.1
user@R2# set route 192.168.40.0/24 next-hop 192.168.10.1
user@R2# set route 192.168.30.0/24 next-hop 192.168.20.1
```

6. Configure the autonomous system (AS) number and the router ID.

```
[edit routing-options]
user@R2# set router-id 2.2.2.2
user@R2# set autonomous-system 2
```

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

```
user@R2# show interfaces
fe-1/2/0 {
    unit 2 {
        family inet {
            address 192.168.10.2/24;
        }
    }
}
fe-1/2/1 {
    unit 3 {
        family inet {
            address 192.168.20.2/24;
        }
    }
}
lo0 {
    unit 2 {
```

```
      family inet {
        address 2.2.2.2/32;
      }
    }
}
```

```
user@R2# show policy-options
policy-statement send-direct {
    term 1 {
        from protocol direct;
        then accept;
    }
}
policy-statement send-local {
    term 1 {
        from protocol local;
        then accept;
    }
}
policy-statement send-static {
    term 1 {
        from protocol static;
        then accept;
    }
}
```

```
user@R2# show protocols
bgp {
    path-selection as-path-ignore;
    group ext {
        type external;
        export [ send-direct send-static send-local ];
        neighbor 192.168.10.1 {
            peer-as 1;
        }
        neighbor 192.168.20.1 {
            peer-as 3;
        }
    }
}
```

```
user@R21# show routing-options
static {
    route 192.168.50.0/24 next-hop 192.168.10.1;
    route 192.168.40.0/24 next-hop 192.168.10.1;
    route 192.168.30.0/24 next-hop 192.168.20.1;
}
router-id 2.2.2.2;
autonomous-system 2;
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat the configuration on the other devices in the network, changing the interface names and IP addresses, as needed.

Verification

Confirm that the configuration is working properly.

- Checking the Neighbor Status on page 207

Checking the Neighbor Status

Purpose Make sure that from Device R2, the active path to get to AS 4 is through AS 1 and AS 5, not through AS 3.

NOTE: To verify the functionality of the **as-path-ignore** statement, you might need to run the **restart routing** command to force reevaluation of the active path. This is because for BGP, if both paths are external, the Junos OS behavior is to prefer the currently active path. This behavior helps to minimize route-flapping. Use caution when restarting the routing protocol process in a production network.

Action From operational mode, enter the **restart routing** command.

```
user@R2> restart routing
Routing protocols process started, pid 49396
```

From operational mode, enter the **show route 4.4.4.4 protocol bgp** command.

```
user@R2> show route 4.4.4.4 protocol bgp
inet.0: 12 destinations, 25 routes (12 active, 0 holddown, 4 hidden)
+ = Active Route, - = Last Active, * = Both

4.4.4.4/32         *[BGP/170] 00:00:12, localpref 100
                      AS path: 1 5 4 I
                    > to 192.168.10.1 via fe-1/2/0.2
                    [BGP/170] 00:00:08, localpref 100
                      AS path: 3 4 I
                    > to 192.168.20.1 via fe-1/2/1.3
```

Meaning The asterisk (*) is next to the path learned from R1, meaning that this is the active path. The AS path for the active path is 1 5 4, which is longer than the AS path (3 4) for the nonactive path learned from Router R3.

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

Example: Removing Private AS Numbers

- Understanding Private AS Number Removal from AS Paths on page 208
- Example: Removing Private AS Numbers from AS Paths on page 209

Understanding Private AS Number Removal from AS Paths

By default, when BGP advertises AS paths to remote systems, it includes all AS numbers, including private AS numbers. You can configure the software so that it removes private AS numbers from AS paths. Doing this is useful when any of the following circumstances are true:

- A remote AS for which you provide connectivity is multihomed, but only to the local AS.
- The remote AS does not have an officially allocated AS number.
- It is not appropriate to make the remote AS a confederation member AS of the local AS.

Most companies acquire their own AS number. Some companies also use private AS numbers to connect to their public AS network. These companies might use a different private AS number for each region in which their company does business. In any implementation, announcing a private AS number to the Internet must be avoided. Service providers can use the remove-private statement to prevent advertising private AS numbers to the Internet.

In an enterprise scenario, suppose that you have multiple AS numbers in your company, some of which are private AS numbers, and one with a public AS number. The one with a public AS number has a direct connection to the service provider. In the AS that connects directly to the service provider, you can use the **remove-private** statement to filter out any private AS numbers in the advertisements that are sent to the service provider.

CAUTION: Changing configuration statements that affect BGP peers, such as enabling or disabling remove-private or renaming a BGP group, resets the BGP sessions. Changes that affect BGP peers should only be made when resetting a BGP session is acceptable.

The AS numbers are stripped from the AS path starting at the left end of the AS path (the end where AS paths have been most recently added). The routing device stops searching for private ASs when it finds the first nonprivate AS or a peer's private AS. If the AS path contains the AS number of the external BGP (EBGP) neighbor, BGP does not remove the private AS number.

NOTE: As of Junos OS 10.0R2 and later, if there is a need to send prefixes to an EBGP peer that has an AS number that matches an AS number in the AS path, consider using the as-override statement instead of the remove-private statement.

The operation takes place after any confederation member ASs have already been removed from the AS path, if applicable.

The software is preconfigured with knowledge of the set of AS numbers that is considered private, a range that is defined in the Internet Assigned Numbers Authority (IANA) assigned numbers document. The set of AS numbers reserved as private are in the range from 64,512 through 65,534, inclusive.

Example: Removing Private AS Numbers from AS Paths

This example demonstrates the removal of a private AS number from the advertised AS path to avoid announcing the private AS number to the Internet.

Requirements

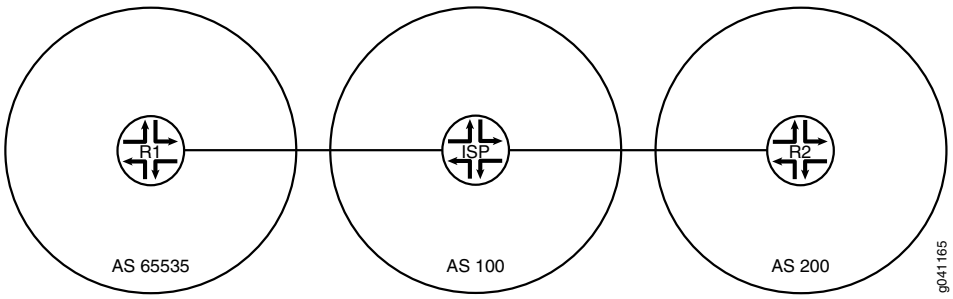
No special configuration beyond device initialization is required before you configure this example.

Overview

Service providers and enterprise networks use the remove-private statement to prevent advertising private AS numbers to the Internet. The **remove-private** statement works in the outbound direction. You configure the **remove-private** statement on a device that has a public AS number and that is connected to one or more devices that have private AS numbers. Generally, you would not configure this statement on a device that has a private AS number.

Figure 21 on page 209 shows the sample topology.

Figure 21: Topology for Removing a Private AS from the Advertised AS Path



In this example, Device R1 is connected to its service provider using private AS number 65535. The example shows the **remove-private** statement configured on Device ISP to prevent Device R1's private AS number from being announced to Device R2. Device R2 sees only the AS number of the service provider.

Configuration

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 192.168.10.1/24
set interfaces lo0 unit 1 family inet address 10.10.10.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext peer-as 100
set protocols bgp group ext neighbor 192.168.10.10
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.20.0/24 next-hop 192.168.10.10
set routing-options autonomous-system 65535
```

Device ISP

```
set interfaces fe-1/2/0 unit 2 family inet address 192.168.10.10/24
set interfaces fe-1/2/1 unit 3 family inet address 192.168.20.20/24
set interfaces lo0 unit 2 family inet address 10.10.0.1/32
set protocols bgp group ext type external
set protocols bgp group ext neighbor 192.168.10.1 peer-as 65535
set protocols bgp group ext neighbor 192.168.20.1 remove-private
set protocols bgp group ext neighbor 192.168.20.1 peer-as 200
set routing-options autonomous-system 100
```

Device R2

```
set interfaces fe-1/2/0 unit 4 family inet address 192.168.20.1/24
set interfaces lo0 unit 3 family inet address 10.10.20.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct
set protocols bgp group ext export send-static
set protocols bgp group ext peer-as 100
set protocols bgp group ext neighbor 192.168.20.20
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options policy-statement send-static term 1 from protocol static
set policy-options policy-statement send-static term 1 then accept
set routing-options static route 192.168.10.0/24 next-hop 192.168.20.20
set routing-options autonomous-system 200
```

Device ISP

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 Device ISP:

1. Configure the interfaces.

   [edit interfaces]

```
user@ISP# set fe-1/2/0 unit 2 family inet address 192.168.10.10/24
user@ISP# set fe-1/2/1 unit 3 family inet address 192.168.20.20/24
user@ISP# set lo0 unit 2 family inet address 10.10.0.1/32
```

2. Configure EBGP.

```
[edit protocols bgp group ext]
user@ISP# set type external
user@ISP# set neighbor 192.168.10.1 peer-as 65535
user@ISP# set neighbor 192.168.20.1 peer-as 200
```

3. For the neighbor in autonomous system (AS) 200 (Device R2), remove private AS numbers from the advertised AS paths.

```
[edit protocols bgp group ext]
user@ISP# set neighbor 192.168.20.1 remove-private
```

4. Configure the AS number.

```
[edit routing-options]
user@ISP# set autonomous-system 100
```

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

```
user@ISP# show interfaces
fe-1/2/0 {
    unit 2 {
        family inet {
            address 192.168.10.10/24;
        }
    }
}
fe-1/2/1 {
    unit 3 {
        family inet {
            address 192.168.20.20/24;
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 10.10.0.1/32;
        }
    }
}
```

```
user@ISP# show protocols
bgp {
    group ext {
        type external;
        neighbor 192.168.10.1 {
            peer-as 65535;
        }
        neighbor 192.168.20.1 {
            remove-private;
```

```
        peer-as 200;
      }
    }
  }
```

```
user@ISP# show routing-options
autonomous-system 100;
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat the configuration on Device R1 and Device R2, changing the interface names and IP address, as needed, and adding the routing policy configuration.

Verification

Confirm that the configuration is working properly.

- Checking the Neighbor Status on page 212
- Checking the Routing Tables on page 213
- Checking the AS Path When the remove-private Statement Is Deactivated on page 213

Checking the Neighbor Status

Purpose Make sure that Device ISP has the **remove-private** setting enabled in its neighbor session with Device R2.

Action From operational mode, enter the **show bgp neighbor 192.168.20.1** command.

```
user@ISP> show bgp neighbor 192.168.20.1
Peer: 192.168.20.1+179 AS 200  Local: 192.168.20.20+60216 AS 100
  Type: External    State: Established    Flags: <ImportEval Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference RemovePrivateAS PeerAS Refresh>
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.10.20.1      Local ID: 10.10.0.1         Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  Local Interface: fe-1/2/1.3
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 200)
  Peer does not support Addpath
  Table inet.0 Bit: 10001
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              1
    Received prefixes:            3
    Accepted prefixes:            2
    Suppressed due to damping:    0
    Advertised prefixes:          1
```

```
  Last traffic (seconds): Received 10   Sent 16   Checked 55
  Input messages:  Total 54     Updates 3       Refreshes 0     Octets 1091
  Output messages: Total 54     Updates 1       Refreshes 0     Octets 1118
  Output Queue[0]: 0
```

Meaning The **RemovePrivateAS** option shows that Device ISP has the expected setting.

Checking the Routing Tables

Purpose Make sure that the devices have the expected routes and AS paths.

Action From operational mode, enter the **show route protocol bgp** command.

```
user@R1> show route protocol bgp
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.20.1/32      *[BGP/170] 00:28:57, localpref 100
                      AS path: 100 200 I
                    > to 192.168.10.10 via fe-1/2/0.1
```

```
user@ISP> show route protocol bgp

inet.0: 7 destinations, 11 routes (7 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.1/32      *[BGP/170] 00:29:40, localpref 100
                      AS path: 65535 I
                    > to 192.168.10.1 via fe-1/2/0.2
10.10.20.1/32      *[BGP/170] 00:29:36, localpref 100
                      AS path: 200 I
                    > to 192.168.20.1 via fe-1/2/1.3
192.168.10.0/24     [BGP/170] 00:29:40, localpref 100
                      AS path: 65535 I
                    > to 192.168.10.1 via fe-1/2/0.2
192.168.20.0/24     [BGP/170] 00:29:36, localpref 100
                      AS path: 200 I
                    > to 192.168.20.1 via fe-1/2/1.3
```

```
user@R2> show route protocol bgp
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.1/32      *[BGP/170] 00:29:53, localpref 100
                      AS path: 100 I
                    > to 192.168.20.20 via fe-1/2/0.4
```

Meaning Device ISP has the private AS number 65535 in its AS path to Device R1. However, Device ISP does not advertise this private AS number to Device R2. This is shown in the routing table of Device R2. Device R2's path to Device R1 contains only the AS number for Device ISP.

Checking the AS Path When the remove-private Statement Is Deactivated

Purpose Verify that without the **remove-private** statement, the private AS number appears in Device R2's routing table.

Action From configuration mode on Device ISP, enter the **deactivate remove-private** command and then recheck the routing table on Device R2.

```
[protocols bgp group ext neighbor 192.168.20.1]
user@ISP# deactivate remove-private
user@ISP# commit
```

```
user@R2> show route protocol bgp
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.1/32      *[BGP/170] 00:00:54, localpref 100
                      AS path: 100 65535 I
                    > to 192.168.20.20 via fe-1/2/0.4
```

Meaning Private AS number 65535 appears in Device R2's AS path to Device R1.

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

CHAPTER 5

BGP BFD Configuration

Example: Configuring BFD for BGP

Understanding BFD for BGP

The Bidirectional Forwarding Detection (BFD) protocol is a simple hello mechanism that detects failures in a network. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. BFD works with a wide variety of network environments and topologies. The failure detection timers for BFD have shorter time limits than default failure detection mechanisms for BGP, so they provide faster detection.

The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the **clear bfd adaptation** command to return BFD interval timers to their configured values. The **clear bfd adaptation** command is hitless, meaning that the command does not affect traffic flow on the routing device.

In Junos OS Release 8.3 and later, BFD is supported on internal BGP (IBGP) and multihop external BGP (EBGP) sessions as well as on single-hop EBGP sessions. In Junos OS Release 9.1 through Junos OS Release 11.1, BFD supports IPv6 interfaces in static routes only. In Junos OS Release 11.2 and later, BFD supports IPv6 interfaces with BGP.

Example: Configuring BFD on Internal BGP Peer Sessions

This example shows how to configure internal BGP (IBGP) peer sessions with the Bidirectional Forwarding Detection (BFD) protocol to detect failures in a network.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

The minimum configuration to enable BFD on IBGP sessions is to include the bfd-liveness-detection **minimum-interval** statement in the BGP configuration of all neighbors participating in the BFD session. The **minimum-interval** statement specifies the minimum transmit and receive intervals for failure detection. Specifically, this value represents the minimum interval after which the local routing device transmits hello packets as well as the minimum interval that the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a value from 1 through 255,000 milliseconds.

Optionally, you can specify the minimum transmit and receive intervals separately using the **transmit-interval minimum-interval** and **minimum-receive-interval** statements. For information about these and other optional BFD configuration statements, see bfd-liveness-detection.

NOTE: BFD is an intensive protocol that consumes system resources. Specifying a minimum interval for BFD less than 100 ms for Routing Engine-based sessions and less than 10 ms for distributed BFD sessions can cause undesired BFD flapping.

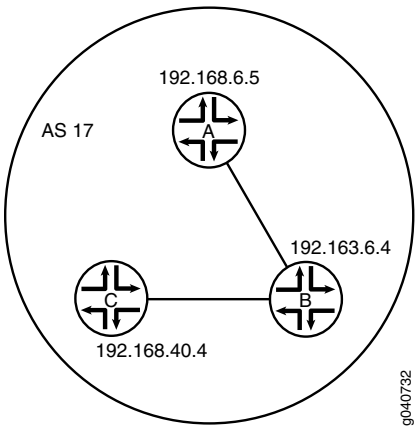
Depending on your network environment, these additional recommendations might apply:

- For large-scale network deployments with a large number of BFD sessions, specify a minimum interval of 300 ms for Routing Engine-based sessions and 100 ms for distributed BFD sessions.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing (NSR) is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with NSR configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

BFD is supported on the default routing instance (the main router), routing instances, and logical systems. This example shows BFD on logical systems.

Figure 22 on page 217 shows a typical network with internal peer sessions.

Figure 22: Typical Network with IBGP Sessions



Configuration

CLI Quick Configuration

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

Device A

```
set logical-systems A interfaces lt-1/2/0 unit 1 description to-B
set logical-systems A interfaces lt-1/2/0 unit 1 encapsulation ethernet
set logical-systems A interfaces lt-1/2/0 unit 1 peer-unit 2
```

```
set logical-systems A interfaces lt-1/2/0 unit 1 family inet address 10.10.10.1/30
set logical-systems A interfaces lo0 unit 1 family inet address 192.168.6.5/32
set logical-systems A protocols bgp group internal-peers type internal
set logical-systems A protocols bgp group internal-peers traceoptions file bgp-bfd
set logical-systems A protocols bgp group internal-peers traceoptions flag bfd detail
set logical-systems A protocols bgp group internal-peers local-address 192.168.6.5
set logical-systems A protocols bgp group internal-peers export send-direct
set logical-systems A protocols bgp group internal-peers bfd-liveness-detection
    minimum-interval 1000
set logical-systems A protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems A protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems A protocols ospf area 0.0.0.0 interface lo0.1 passive
set logical-systems A protocols ospf area 0.0.0.0 interface lt-1/2/0.1
set logical-systems A policy-options policy-statement send-direct term 2 from protocol
    direct
set logical-systems A policy-options policy-statement send-direct term 2 then accept
set logical-systems A routing-options router-id 192.168.6.5
set logical-systems A routing-options autonomous-system 17
```

Device B

```
set logical-systems B interfaces lt-1/2/0 unit 2 description to-A
set logical-systems B interfaces lt-1/2/0 unit 2 encapsulation ethernet
set logical-systems B interfaces lt-1/2/0 unit 2 peer-unit 1
set logical-systems B interfaces lt-1/2/0 unit 2 family inet address 10.10.10.2/30
set logical-systems B interfaces lt-1/2/0 unit 5 description to-C
set logical-systems B interfaces lt-1/2/0 unit 5 encapsulation ethernet
set logical-systems B interfaces lt-1/2/0 unit 5 peer-unit 6
set logical-systems B interfaces lt-1/2/0 unit 5 family inet address 10.10.10.5/30
set logical-systems B interfaces lo0 unit 2 family inet address 192.163.6.4/32
set logical-systems B protocols bgp group internal-peers type internal
set logical-systems B protocols bgp group internal-peers local-address 192.163.6.4
set logical-systems B protocols bgp group internal-peers export send-direct
set logical-systems B protocols bgp group internal-peers bfd-liveness-detection
    minimum-interval 1000
set logical-systems B protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems B protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems B protocols ospf area 0.0.0.0 interface lo0.2 passive
set logical-systems B protocols ospf area 0.0.0.0 interface lt-1/2/0.2
set logical-systems B protocols ospf area 0.0.0.0 interface lt-1/2/0.5
set logical-systems B policy-options policy-statement send-direct term 2 from protocol
    direct
set logical-systems B policy-options policy-statement send-direct term 2 then accept
set logical-systems B routing-options router-id 192.163.6.4
set logical-systems B routing-options autonomous-system 17
```

Device C

```
set logical-systems C interfaces lt-1/2/0 unit 6 description to-B
set logical-systems C interfaces lt-1/2/0 unit 6 encapsulation ethernet
set logical-systems C interfaces lt-1/2/0 unit 6 peer-unit 5
set logical-systems C interfaces lt-1/2/0 unit 6 family inet address 10.10.10.6/30
set logical-systems C interfaces lo0 unit 3 family inet address 192.168.40.4/32
set logical-systems C protocols bgp group internal-peers type internal
set logical-systems C protocols bgp group internal-peers local-address 192.168.40.4
set logical-systems C protocols bgp group internal-peers export send-direct
set logical-systems C protocols bgp group internal-peers bfd-liveness-detection
    minimum-interval 1000
set logical-systems C protocols bgp group internal-peers neighbor 192.163.6.4
```

```
set logical-systems C protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems C protocols ospf area 0.0.0.0 interface lo0.3 passive
set logical-systems C protocols ospf area 0.0.0.0 interface lt-1/2/0.6
set logical-systems C policy-options policy-statement send-direct term 2 from protocol
  direct
set logical-systems C policy-options policy-statement send-direct term 2 then accept
set logical-systems C routing-options router-id 192.168.40.4
set logical-systems C routing-options autonomous-system 17
```

Configuring Device A

Step-by-Step Procedure

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

To configure Device A:

1. Set the CLI to Logical System A.

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

2. Configure the interfaces.

   ```
   [edit interfaces lt-1/2/0 unit 1]
   user@host:A# set description to-B
   user@host:A# set encapsulation ethernet
   user@host:A# set peer-unit 2
   user@host:A# set family inet address 10.10.10.1/30

   [edit interfaces lo0 unit 1]
   user@host:A# set family inet address 192.168.6.5/32
   ```

3. Configure BGP.

   The **neighbor** statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

   ```
   [edit protocols bgp group internal-peers]
   user@host:A# set type internal
   user@host:A# set local-address 192.168.6.5
   user@host:A# set export send-direct
   user@host:A# set neighbor 192.163.6.4
   user@host:A# set neighbor 192.168.40.4
   ```

4. Configure BFD.

   ```
   [edit protocols bgp group internal-peers]
   user@host:A# set bfd-liveness-detection minimum-interval 1000
   ```

   You must configure the same minimum interval on the connecting peer.

5. (Optional) Configure BFD tracing.

   ```
   [edit protocols bgp group internal-peers]
   user@host:A# set traceoptions file bgp-bfd
   user@host:A# set traceoptions flag bfd detail
   ```

6. Configure OSPF.

   ```
   [edit protocols ospf area 0.0.0.0]
   ```

```
user@host:A# set interface lo0.1 passive
user@host:A# set interface lt-1/2/0.1
```

7. Configure a policy that accepts direct routes.

   Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

   ```
   [edit policy-options policy-statement send-direct term 2]
   user@host:A# set from protocol direct
   user@host:A# set then accept
   ```

8. Configure the router ID and the autonomous system (AS) number.

   ```
   [edit routing-options]
   user@host:A# set router-id 192.168.6.5
   user@host:A# set autonomous-system 17
   ```

9. If you are done configuring the device, enter **commit** from configuration mode. Repeat these steps to configure Device B and Device C.

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

```
user@host:A# show interfaces
lt-1/2/0 {
  unit 1 {
    description to-B;
    encapsulation ethernet;
    peer-unit 2;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.6.5/32;
    }
  }
}
```

```
user@host:A# show policy-options
policy-statement send-direct {
  term 2 {
    from protocol direct;
    then accept;
  }
}
```

```
user@host:A# show protocols
bgp {
  group internal-peers {
    type internal;
    traceoptions {
```



```
        file bgp-bfd;
        flag bfd detail;
      }
      local-address 192.168.6.5;
      export send-direct;
      bfd-liveness-detection {
        minimum-interval 1000;
      }
      neighbor 192.163.6.4;
      neighbor 192.168.40.4;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.1 {
        passive;
      }
      interface lt-1/2/0.1;
    }
  }
```

```
user@host:A# show routing-options
router-id 192.168.6.5;
autonomous-system 17;
```

Verification

Confirm that the configuration is working properly.

- Verifying That BFD Is Enabled on page 221
- Verifying That BFD Sessions Are Up on page 222
- Viewing Detailed BFD Events on page 222
- Viewing Detailed BFD Events After Deactivating and Reactivating a Loopback Interface on page 223

Verifying That BFD Is Enabled

Purpose Verify that BFD is enabled between the IBGP peers.

Action From operational mode, enter the **show bgp neighbor** command. You can use the **| match bfd** filter to narrow the output.

```
user@host:A> show bgp neighbor | match bfd
Options: <BfdEnabled>
  BFD: enabled, up
  Trace file: /var/log/A/bgp-bfd size 131072 files 10
  Options: <BfdEnabled>
  BFD: enabled, up
  Trace file: /var/log/A/bgp-bfd size 131072 files 10
```

Meaning The output shows that Logical System A has two neighbors with BFD enabled. When BFD is not enabled, the output displays **BFD: disabled, down**, and the **<BfdEnabled>** option is absent. If BFD is enabled and the session is down, the output displays **BFD: enabled,**

down. The output also shows that BFD-related events are being written to a log file because trace operations are configured.

Verifying That BFD Sessions Are Up

Purpose Verify that the BFD sessions are up, and view details about the BFD sessions.

Action From operational mode, enter the **show bfd session extensive** command.

```
user@host:A> show bfd session extensive
                                                  Detect   Transmit
Address                 State     Interface       Time     Interval  Multiplier
192.163.6.4             Up                        3.000     1.000        3
 Client BGP, TX interval 1.000, RX interval 1.000
 Session up time 00:54:40
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Logical system 12, routing table index 25
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
 Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
 Local discriminator 10, remote discriminator 9
 Echo mode disabled/inactive
 Multi-hop route table 25, local-address 192.168.6.5

                                                  Detect   Transmit
Address                 State     Interface       Time     Interval  Multiplier
192.168.40.4            Up                        3.000     1.000        3
 Client BGP, TX interval 1.000, RX interval 1.000
 Session up time 00:48:03
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Logical system 12, routing table index 25
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
 Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
 Local discriminator 14, remote discriminator 13
 Echo mode disabled/inactive
 Multi-hop route table 25, local-address 192.168.6.5

2 sessions, 2 clients
Cumulative transmit rate 2.0 pps, cumulative receive rate 2.0 pps
```

Meaning The **TX interval 1.000, RX interval 1.000** output represents the setting configured with the **minimum-interval** statement. All of the other output represents the default settings for BFD. To modify the default settings, include the optional statements under the bfd-liveness-detection statement.

Viewing Detailed BFD Events

Purpose View the contents of the BFD trace file to assist in troubleshooting, if needed.

Action From operational mode, enter the **file show /var/log/A/bgp-bfd** command.

```
user@host:A> file show /var/log/A/bgp-bfd
```

```
Aug 15 17:07:25 trace_on: Tracing to "/var/log/A/bgp-bfd" started
Aug 15 17:07:26.492190 bgp_peer_init: BGP peer 192.163.6.4 (Internal AS 17) local
 address 192.168.6.5 not found.  Leaving peer idled
Aug 15 17:07:26.493176 bgp_peer_init: BGP peer 192.168.40.4 (Internal AS 17) local
 address 192.168.6.5 not found.  Leaving peer idled
Aug 15 17:07:32.597979 task_connect: task BGP_17.192.163.6.4+179 addr
192.163.6.4+179: No route to host
Aug 15 17:07:32.599623 bgp_connect_start: connect 192.163.6.4 (Internal AS 17):
No route to host
Aug 15 17:07:36.869394 task_connect: task BGP_17.192.168.40.4+179 addr
192.168.40.4+179: No route to host
Aug 15 17:07:36.870624 bgp_connect_start: connect 192.168.40.4 (Internal AS 17):
 No route to host
Aug 15 17:08:04.599220 task_connect: task BGP_17.192.163.6.4+179 addr
192.163.6.4+179: No route to host
Aug 15 17:08:04.601135 bgp_connect_start: connect 192.163.6.4 (Internal AS 17):
No route to host
Aug 15 17:08:08.869717 task_connect: task BGP_17.192.168.40.4+179 addr
192.168.40.4+179: No route to host
Aug 15 17:08:08.869934 bgp_connect_start: connect 192.168.40.4 (Internal AS 17):
 No route to host
Aug 15 17:08:36.603544 advertising receiving-speaker only capabilty to neighbor
192.163.6.4 (Internal AS 17)
Aug 15 17:08:36.606726 bgp_read_message: 192.163.6.4 (Internal AS 17): 0 bytes
buffered
Aug 15 17:08:36.609119 Initiated BFD session to peer 192.163.6.4 (Internal AS
17): address=192.163.6.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3
ver=255
Aug 15 17:08:36.734033 advertising receiving-speaker only capabilty to neighbor
192.168.40.4 (Internal AS 17)
Aug 15 17:08:36.738436 Initiated BFD session to peer 192.168.40.4 (Internal AS
17): address=192.168.40.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3
ver=255
Aug 15 17:08:40.537552 BFD session to peer 192.163.6.4 (Internal AS 17) up
Aug 15 17:08:40.694410 BFD session to peer 192.168.40.4 (Internal AS 17) up
```

Meaning Before the routes are established, the **No route to host** message appears in the output. After the routes are established, the last two lines show that both BFD sessions come up.

Viewing Detailed BFD Events After Deactivating and Reactivating a Loopback Interface

Purpose Check to see what happens after bringing down a router or switch and then bringing it back up. To simulate bringing down a router or switch, deactivate the loopback interface on Logical System B.

Action

1. From configuration mode, enter the **deactivate logical-systems B interfaces lo0 unit 2 family inet** command.

   ```
   user@host:A# deactivate logical-systems B interfaces lo0 unit 2 family inet
   user@host:A# commit
   ```

2. From operational mode, enter the **file show /var/log/A/bgp-bfd** command.

   ```
   user@host:A> file show /var/log/A/bgp-bfd
   ...
   Aug 15 17:20:55.995648 bgp_read_v4_message:9747: NOTIFICATION received from
   192.163.6.4 (Internal AS 17): code 6 (Cease) subcode 6 (Other Configuration
   Change)
   ```



```
Aug 15 17:20:56.004508 Terminated BFD session to peer 192.163.6.4 (Internal
AS 17)
Aug 15 17:21:28.007755 task_connect: task BGP_17.192.163.6.4+179 addr
192.163.6.4+179: No route to host
Aug 15 17:21:28.008597 bgp_connect_start: connect 192.163.6.4 (Internal AS
17): No route to host
```

3. From configuration mode, enter the **activate logical-systems B interfaces lo0 unit 2 family inet** command.

```
user@host:A# activate logical-systems B interfaces lo0 unit 2 family inet
user@host:A# commit
```

4. From operational mode, enter the **file show /var/log/A/bgp-bfd** command.

```
user@host:A> file show /var/log/A/bgp-bfd
...
Aug 15 17:25:53.623743 advertising receiving-speaker only capabilty to neighbor
 192.163.6.4 (Internal AS 17)
Aug 15 17:25:53.631314 Initiated BFD session to peer 192.163.6.4 (Internal AS
 17): address=192.163.6.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3
 ver=255
Aug 15 17:25:57.570932 BFD session to peer 192.163.6.4 (Internal AS 17) up
```

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

Example: Configuring BFD Authentication for BGP

- Understanding BFD Authentication for BGP on page 224
- Example: Configuring BFD Authentication for BGP on page 226

Understanding BFD Authentication for BGP

Bidirectional Forwarding Detection protocol (BFD) enables rapid detection of communication failures between adjacent systems. By default, authentication for BFD sessions is disabled. However, when you run BFD over Network Layer protocols, the risk of service attacks can be significant. We strongly recommend using authentication if you are running BFD over multiple hops or through insecure tunnels. Beginning with Junos OS Release 9.6, Junos OS supports authentication for BFD sessions running over BGP. BFD authentication is not supported on MPLS OAM sessions. BFD authentication is only supported in the Canada and United States version of the Junos OS image and is not available in the export version.

You authenticate BFD sessions by specifying an authentication algorithm and keychain, and then associating that configuration information with a security authentication keychain using the keychain name.

The following sections describe the supported authentication algorithms, security keychains, and level of authentication that can be configured:

BFD Authentication Algorithms

Junos OS supports the following algorithms for BFD authentication:

- **simple-password**—Plain-text password. One to 16 bytes of plain text are used to authenticate the BFD session. One or more passwords can be configured. This method is the least secure and should be used only when BFD sessions are not subject to packet interception.
- **keyed-md5**—Keyed Message Digest 5 hash algorithm for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed MD5 uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than or equal to the last sequence number received. Although more secure than a simple password, this method is vulnerable to replay attacks. Increasing the rate at which the sequence number is updated can reduce this risk.
- **meticulous-keyed-md5**—Meticulous keyed Message Digest 5 hash algorithm. This method works in the same manner as keyed MD5, but the sequence number is updated with every packet. Although more secure than keyed MD5 and simple passwords, this method might take additional time to authenticate the session.
- **keyed-sha-1**—Keyed Secure Hash Algorithm I for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed SHA uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. The key is not carried within the packets. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than the last sequence number received.
- **meticulous-keyed-sha-1**—Meticulous keyed Secure Hash Algorithm I. This method works in the same manner as keyed SHA, but the sequence number is updated with every packet. Although more secure than keyed SHA and simple passwords, this method might take additional time to authenticate the session.

NOTE: Nonstop active routing (NSR) is not supported with meticulous-keyed-md5 and meticulous-keyed-sha-1 authentication algorithms. BFD sessions using these algorithms might go down after a switchover.

Security Authentication Keychains

The security authentication keychain defines the authentication attributes used for authentication key updates. When the security authentication keychain is configured and associated with a protocol through the keychain name, authentication key updates can occur without interrupting routing and signaling protocols.

The authentication keychain contains one or more keychains. Each keychain contains one or more keys. Each key holds the secret data and the time at which the key becomes valid. The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

BFD allows multiple clients per session, and each client can have its own keychain and algorithm defined. To avoid confusion, we recommend specifying only one security authentication keychain.

Strict Versus Loose Authentication

By default, strict authentication is enabled and authentication is checked at both ends of each BFD session. Optionally, to smooth migration from nonauthenticated sessions to authenticated sessions, you can configure *loose checking*. When loose checking is configured, packets are accepted without authentication being checked at each end of the session. This feature is intended for transitional periods only.

Example: Configuring BFD Authentication for BGP

Beginning with Junos OS Release 9.6, you can configure authentication for BFD sessions running over BGP. Only three steps are needed to configure authentication on a BFD session:

1. Specify the BFD authentication algorithm for the BGP protocol.
2. Associate the authentication keychain with the BGP protocol.
3. Configure the related security authentication keychain.

The following sections provide instructions for configuring and viewing BFD authentication on BGP:

- Configuring BFD Authentication Parameters on page 226
- Viewing Authentication Information for BFD Sessions on page 228

Configuring BFD Authentication Parameters

BFD authentication can be configured for the entire BGP protocol, or a specific BGP group, neighbor, or routing instance.

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 BFD authentication:

1. Specify the algorithm (**keyed-md5**, **keyed-sha-1**, **meticulous-keyed-md5**, **meticulous-keyed-sha-1**, or **simple-password**) to use.

   ```
   [edit]
   user@host# set protocols bgp bfd-liveness-detection authentication algorithm
     keyed-sha-1
   user@host# set protocols bgp group bgp-gr1 bfd-liveness-detection authentication
     algorithm keyed-sha-1
   user@host# set protocols bgp group bgp-gr1 neighbor 10.10.10.7 bfd-liveness-detection
     authentication algorithm keyed-sha-1
   ```

   NOTE: Nonstop active routing is not supported with meticulous-keyed-md5 and meticulous-keyed-sha-1 authentication algorithms. BFD sessions using these algorithms might go down after a switchover.

2. Specify the keychain to be used to associate BFD sessions on BGP with the unique security authentication keychain attributes.

   The keychain name you specify must match a keychain name configured at the **[edit security authentication key-chains]** hierarchy level.

   ```
   [edit]
   user@host# set protocols bgp bfd-liveness-detection authentication keychain bfd-bgp
   user@host# set protocols bgp group bgp-gr1 bfd-liveness-detection authentication
     keychain bfd-bgp
   user@host# set protocols bgp group bgp-gr1 neighbor 10.10.10.7 bfd-liveness-detection
     authentication keychain bfd-bgp
   ```

   NOTE: The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

3. Specify the unique security authentication information for BFD sessions:

   - The matching keychain name as specified in Step 2.
   - At least one key, a unique integer between **0** and **63**. Creating multiple keys allows multiple clients to use the BFD session.
   - The secret data used to allow access to the session.
   - The time at which the authentication key becomes active, in the format *yyyy-mm-dd.hh:mm:ss*.

     ```
     [edit security]
     ```

```
user@host# set authentication-key-chains key-chain bfd-bgp key 53 secret
  $9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm start-time 2009-06-14.10:00:00
```

4. (Optional) Specify loose authentication checking if you are transitioning from nonauthenticated sessions to authenticated sessions.

```
[edit]
user@host# set protocols bgp bfd-liveness-detection authentication loose-check
user@host# set protocols bgp group bgp-gr1 bfd-liveness-detection authentication
  loose-check
user@host# set protocols bgp group bgp-gr1 neighbor 10.10.10.7 bfd-liveness-detection
  authentication loose-check
```

5. (Optional) View your configuration using the **show bfd session detail** or **show bfd session extensive** command.

6. Repeat these steps to configure the other end of the BFD session.

NOTE: BFD authentication is only supported in the Canada and United States version of the Junos OS image and is not available in the export version.

Viewing Authentication Information for BFD Sessions

You can view the existing BFD authentication configuration using the **show bfd session detail** and **show bfd session extensive** commands.

The following example shows BFD authentication configured for the **bgp-gr1** BGP group. It specifies the keyed SHA-1 authentication algorithm and a keychain name of **bfd-bgp**. The authentication keychain is configured with two keys. Key **1** contains the secret data "**9ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm**" and a start time of June 1, 2009, at 9:46:02 AM PST. Key **2** contains the secret data "**9a5jiKW9l.reP38ny.TszF2/9**" and a start time of June 1, 2009, at 3:29:20 PM PST.

```
[edit protocols bgp]
group bgp-gr1 {
   bfd-liveness-detection {
      authentication {
         algorithm keyed-sha-1;
         key-chain bfd-bgp;
      }
   }
}
[edit security]
authentication key-chains {
   key-chain bfd-bgp {
      key 1 {
         secret "$9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm";
         start-time "2009-6-1.09:46:02 -0700";
      }
      key 2 {
         secret "$9$a5jiKW9l.reP38ny.TszF2/9";
         start-time "2009-6-1.15:29:20 -0700";
      }
   }
```

```
}
```

If you commit these updates to your configuration, you see output similar to the following. In the output for the **show bfd session detail** command, **Authenticate** is displayed to indicate that BFD authentication is configured. For more information about the configuration, use the **show bfd session extensive** command. The output for this command provides the keychain name, the authentication algorithm and mode for each client in the session, and the overall BFD authentication configuration status, keychain name, and authentication algorithm and mode.

show bfd session detail

```
user@host# show bfd session detail

                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
50.0.0.2                Up        ge-0/1/5.0    0.900     0.300        3
 Client BGP, TX interval 0.300, RX interval 0.300, Authenticate
 Session up time 3d 00:34
 Local diagnostic None, remote diagnostic NbrSignal
 Remote state Up, version 1
 Replicated
```

show bfd session extensive

```
user@host# show bfd session extensive
                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
50.0.0.2                Up        ge-0/1/5.0    0.900     0.300        3
 Client BGP, TX interval 0.300, RX interval 0.300, Authenticate
        keychain bfd-bgp, algo keyed-sha-1, mode strict
 Session up time 00:04:42
 Local diagnostic None, remote diagnostic NbrSignal
 Remote state Up, version 1
 Replicated
 Min async interval 0.300, min slow interval 1.000
 Adaptive async TX interval 0.300, RX interval 0.300
 Local min TX interval 0.300, minimum RX interval 0.300, multiplier 3
 Remote min TX interval 0.300, min RX interval 0.300, multiplier 3
 Local discriminator 2, remote discriminator 2
 Echo mode disabled/inactive
 Authentication enabled/active, keychain bfd-bgp, algo keyed-sha-1, mode strict
```

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

CHAPTER 6

BGP Load Balancing Configuration

Examples: Configuring BGP Multipath

Understanding BGP Multipath

The Junos OS BGP multipath feature supports the following applications:

- Load balancing across multiple links between two routing devices belonging to different autonomous systems (ASs)
- Load balancing across a common subnet or multiple subnets to different routing devices belonging to the same peer AS
- Load balancing across multiple links between two routing devices belonging to different external confederation peers
- Load balancing across a common subnet or multiple subnets to different routing devices belonging to external confederation peers

In a common scenario for load balancing, a customer is multihomed to multiple routers in a point of presence (POP). The default behavior is to send all traffic across only one of the available links. Load balancing causes traffic to use two or more of the links.

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.

Example: Load Balancing BGP Traffic

This example shows how to configure BGP to select multiple equal-cost external BGP (EBGP) or internal BGP (IBGP) paths as active paths.

Requirements

Before you begin:

- Configure the device interfaces.
- Configure an interior gateway protocol (IGP).
- Configure BGP.
- Configure a routing policy that exports routes (such as direct routes or IGP routes) from the routing table into BGP.

Overview

The following steps show how to configure per-packet load balancing:

1. Define a load-balancing routing policy by including one or more **policy-statement** statements at the **[edit policy-options]** hierarchy level, defining an action of **load-balance per-packet**:

   ```
   policy-statement policy-name {
     from {
       match-conditions;
       route-filter destination-prefix match-type <actions>;
       prefix-list name;
     }
     then {
       load-balance per-packet;
     }
   }
   ```

2. Apply the policy to routes exported from the routing table to the forwarding table. To do this, include the **forwarding-table** and **export** statements:

   ```
   forwarding-table {
     export policy-name;
   }
   ```

   You cannot apply the export policy to VRF routing instances.

3. Specify all next hops of that route, if more than one exists, when allocating a label corresponding to a route that is being advertised.

4. Configure the forwarding-options hash key for MPLS to include the IP payload.

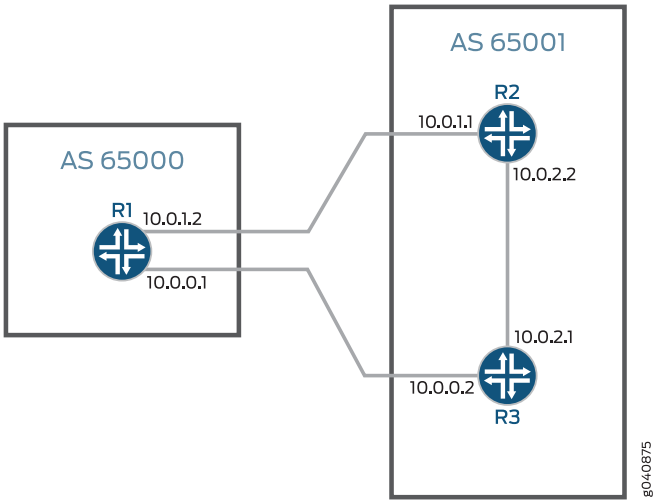
NOTE: On some platforms, you can increase the number of paths that are load balanced by using the **chassis maximum-ecmp** statement. With this statement, you can change the maximum number of equal-cost load-balanced paths to 32 or 64.

In this example, Device R1 is in AS 65000 and is connected to both Device R2 and Device R3, which are in AS 65001. This example shows the configuration on Device R1.

Topology

Figure 23 on page 233 shows the topology used in this example.

Figure 23: BGP Load Balancing



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 protocols bgp group external type external
set protocols bgp group external peer-as 65001
set protocols bgp group external multipath
set protocols bgp group external neighbor 10.0.1.1
set protocols bgp group external neighbor 10.0.0.2
set policy-options policy-statement loadbal from route-filter 10.0.0.0/16 orlonger
set policy-options policy-statement loadbal then load-balance per-packet
set routing-options forwarding-table export loadbal
set routing-options autonomous-system 65000
```

Step-by-Step Procedure

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

To configure the BGP peer sessions:

1. Configure the BGP group.

```
[edit protocols bgp group external]
user@R1# set type external
user@R1# set peer-as 65001
user@R1# set neighbor 10.0.1.1
user@R1# set neighbor 10.0.0.2
```

2. Enable the BGP group to use multiple paths.

NOTE: To disable the default check requiring that paths accepted by BGP multipath must have the same neighboring autonomous system (AS), include the **multiple-as** option.

```
[edit protocols bgp group external]
user@R1# set multipath
```

3. Configure the load-balancing policy.

```
[edit policy-options policy-statement loadbal]
user@R1# set from route-filter 10.0.0.0/16 orlonger
user@R1# set then load-balance per-packet
```

4. Apply the load-balancing policy.

```
[edit routing-options]
user@R1# set forwarding-table export loadbal
```

5. Configure the local autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 65000
```

Results

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

```
[edit]
user@R1# show protocols
bgp {
  group external {
    type external;
    peer-as 65001;
    multipath;
    neighbor 10.0.1.1;
    neighbor 10.0.0.2;
  }
}
```

```
[edit]
user@R1# show policy-options
policy-statement loadbal {
  from {
    route-filter 10.0.0.0/16 orlonger;
  }
  then {
    load-balance per-packet;
  }
}
```

```
[edit]
user@R1# show routing-options
autonomous-system 65000;
forwarding-table {
  export loadbal;
}
```

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

Verification

Confirm that the configuration is working properly:

- Verifying Routes on page 235
- Verifying Forwarding on page 236

Verifying Routes

Purpose Verify that routes are learned from both routers in the neighboring AS.

Action From operational mode, run the **show route** command.

```
user@R1> show route 10.0.2.0
inet.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.2.0/30        *[BGP/170] 03:12:32, localpref 100
                      AS path: 65001 I
                      to 10.0.1.1 via ge-1/2/0.0
                    > to 10.0.0.2 via ge-1/2/1.0
                    [BGP/170] 03:12:32, localpref 100
                      AS path: 65001 I
                    > to 10.0.1.1 via ge-1/2/0.0
```

```
user@R1> show route 10.0.2.0 detail
inet.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
10.0.2.0/30 (2 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Router, Next hop index: 262142
                Next-hop reference count: 3
                Source: 10.0.0.2
                Next hop: 10.0.1.1 via ge-1/2/0.0
                Next hop: 10.0.0.2 via ge-1/2/1.0, selected
                State: <Active Ext>
                Local AS: 65000 Peer AS: 65001
                Age: 3:18:30
                Task: BGP_65001.10.0.0.2+55402
```

```
                Announcement bits (1): 2-KRT
                AS path: 65001 I
                Accepted Multipath
                Localpref: 100
                Router ID: 192.168.2.1
BGP             Preference: 170/-101
                Next hop type: Router, Next hop index: 602
                Next-hop reference count: 5
                Source: 10.0.1.1
                Next hop: 10.0.1.1 via ge-1/2/0.0, selected
                State: <NotBest Ext>
                Inactive reason: Not Best in its group - Active preferred
                Local AS: 65000 Peer AS: 65001
                Age: 3:18:30
                Task: BGP_65001.10.0.1.1+53135
                AS path: 65001 I
                Accepted
                Localpref: 100
                Router ID: 192.168.3.1
```

Meaning The active path, denoted with an asterisk (*), has two next hops: 10.0.1.1 and 10.0.0.2 to the 10.0.2.0 destination. The 10.0.1.1 next hop is copied from the inactive path to the active path.

NOTE: The **show route detail** command output designates one gateway as **selected**. This output is potentially confusing in the context of load balancing. The selected gateway is used for many purposes in addition to deciding which gateway to install into the kernel when Junos OS is not performing per-packet load-balancing. For instance, the **ping mpls** command uses the selected gateway when sending packets. Multicast protocols use the selected gateway in some cases to determine the upstream interface. Therefore, even when Junos OS is performing per-packet load-balancing by way of a forwarding-table policy, the selected gateway information is still required for other purposes. It is useful to display the selected gateway for troubleshooting purposes. Additionally, it is possible to use forwarding-table policy to override what is installed into the kernel (for example, by using the **install-nexthop** action). In this case, the next-hop gateway installed in the forwarding table might be a subset of the total gateways displayed in the **show route** command.

Verifying Forwarding

Purpose Verify that both next hops are installed in the forwarding table.

Action From operational mode, run the **show route forwarding-table** command.

```
user@R1> show route forwarding-table destination 10.0.2.0
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
10.0.2.0/30        user     0                    ulst 262142     2
```

```
10.0.1.1           ucst   602     5 ge-1/2/0.0
10.0.0.2           ucst   522     6 ge-1/2/1.0
```

Example: Configuring Single-Hop EBGP Peers to Accept Remote Next Hops

This example shows how to configure a single-hop external BGP (EBGP) peer to accept a remote next hop with which it does not share a common subnet.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

In some situations, it is necessary to configure a single-hop EBGP peer to accept a remote next hop with which it does not share a common subnet. The default behavior is for any next-hop address received from a single-hop EBGP peer that is not recognized as sharing a common subnet to be discarded. The ability to have a single-hop EBGP peer accept a remote next hop to which it is not directly connected also prevents you from having to configure the single-hop EBGP neighbor as a multihop session. When you configure a multihop session in this situation, all next-hop routes learned through this EBGP peer are labeled indirect even when they do share a common subnet. This situation breaks multipath functionality for routes that are recursively resolved over routes that include these next-hop addresses. Configuring the accept-remote-nexthop statement allows a single-hop EBGP peer to accept a remote next hop, which restores multipath functionality for routes that are resolved over these next-hop addresses. You can configure this statement at the global, group, and neighbor hierarchy levels for BGP. The statement is also supported on logical systems and the VPN routing and forwarding (VRF) routing instance type. Both the remote next-hop and the EBGP peer must support BGP route refresh as defined in RFC 2918, *Route Refresh Capability in BGP-4*. If the remote peer does not support BGP route refresh, the session is reset.

NOTE: **You cannot configure both the multihop and accept-remote-nexthop statements for the same EBGP peer.**

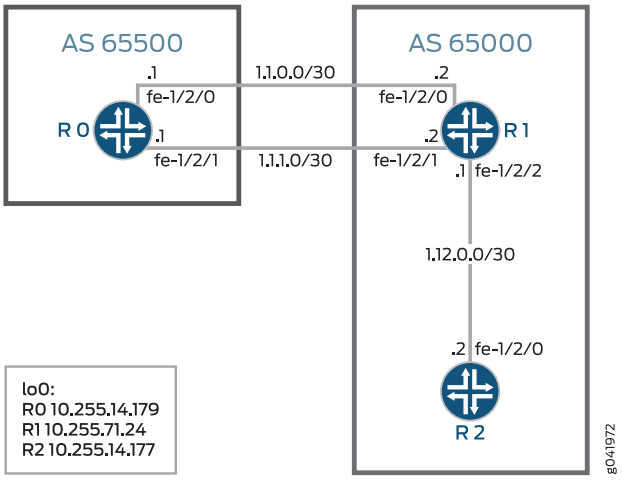
When you enable a single-hop EBGP peer to accept a remote next hop, you must also configure an import routing policy on the EBGP peer that specifies the remote next-hop address.

This example includes an import routing policy, **agg_route**, that enables a single-hop external BGP peer (Device R1) to accept the remote next-hop 1.1.10.10 for the route to the 1.1.230.0/23 network. At the **[edit protocols bgp]** hierarchy level, the example includes the **import agg_route** statement to apply the policy to the external BGP peer and includes

the **accept-remote-nexthop** statement to enable the single-hop EBGP peer to accept the remote next hop.

Figure 24 on page 238 shows the sample topology.

Figure 24: Topology for Accepting a Remote Next Hop



Configuration

- Device R0 on page 239
- Configuring Device R1 on page 242
- Configuring Device R2 on page 244

CLI Quick Configuration

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

Device R0

```
set interfaces fe-1/2/0 unit 1 family inet address 1.1.0.1/30
set interfaces fe-1/2/1 unit 2 family inet address 1.1.1.1/30
set interfaces lo0 unit 1 family inet address 10.255.14.179/32
set protocols bgp group ext type external
set protocols bgp group ext export test_route
set protocols bgp group ext export agg_route
set protocols bgp group ext peer-as 65000
set protocols bgp group ext multipath
set protocols bgp group ext neighbor 1.1.0.2
set protocols bgp group ext neighbor 1.1.1.2
set policy-options policy-statement agg_route term 1 from protocol static
set policy-options policy-statement agg_route term 1 from route-filter 1.1.230.0/23 exact
set policy-options policy-statement agg_route term 1 then accept
set policy-options policy-statement test_route term 1 from protocol static
set policy-options policy-statement test_route term 1 from route-filter 1.1.10.10/32 exact
set policy-options policy-statement test_route term 1 then accept
set routing-options static route 1.1.10.10/32 reject
set routing-options static route 1.1.230.0/23 reject
set routing-options autonomous-system 65500
```

Device R1

```
set interfaces fe-1/2/0 unit 3 family inet address 1.1.0.2/30
set interfaces fe-1/2/1 unit 4 family inet address 1.12.0.1/30
set interfaces fe-1/2/2 unit 5 family inet address 1.1.1.2/30
set interfaces lo0 unit 2 family inet address 10.255.71.24/32
set protocols bgp accept-remote-nexthop
set protocols bgp group ext type external
set protocols bgp group ext import agg_route
set protocols bgp group ext peer-as 65500
set protocols bgp group ext multipath
set protocols bgp group ext neighbor 1.1.0.1
set protocols bgp group ext neighbor 1.1.1.1
set protocols bgp group int type internal
set protocols bgp group int local-address 10.255.71.24
set protocols bgp group int neighbor 10.255.14.177
set protocols ospf area 0.0.0.0 interface fe-1/2/1.4
set protocols ospf area 0.0.0.0 interface 10.255.71.24
set policy-options policy-statement agg_route term 1 from protocol bgp
set policy-options policy-statement agg_route term 1 from route-filter 1.1.230.0/23 exact
set policy-options policy-statement agg_route term 1 then next-hop 1.1.10.10
set policy-options policy-statement agg_route term 1 then accept
set routing-options autonomous-system 65000
```

Device R2

```
set interfaces fe-1/2/0 unit 6 family inet address 1.12.0.2/30
set interfaces lo0 unit 3 family inet address 10.255.14.177/32
set protocols bgp group int type internal
set protocols bgp group int local-address 10.255.14.177
set protocols bgp group int neighbor 10.255.71.24
set protocols ospf area 0.0.0.0 interface fe-1/2/0.6
set protocols ospf area 0.0.0.0 interface 10.255.14.177
set routing-options autonomous-system 65000
```

Device R0

Step-by-Step Procedure

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

To configure Device R0:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 1]
user@R0# set family inet address 1.1.0.1/30

[edit interfaces fe-1/2/1 unit 2]
user@R0# set family inet address 1.1.1.1/30

[edit interfaces lo0 unit 1]
user@R0# set family inet address 10.255.14.179/32
```

2. Configure EBGP.

```
[edit protocols bgp group ext]
user@R0# set type external
user@R0# set peer-as 65000
user@R0# set neighbor 1.1.0.2
```

```
user@R0# set neighbor 1.1.1.2
```

3. Enable multipath BGP between Device R0 and Device R1.

```
[edit protocols bgp group ext]
user@R0# set multipath
```

4. Configure static routes to remote networks.
   These routes are not part of the topology. The purpose of these routes is to demonstrate the functionality in this example.

```
[edit routing-options]
user@R0# set static route 1.1.10.10/32 reject
user@R0# set static route 1.1.230.0/23 reject
```

5. Configure routing policies that accept the static routes.

```
[edit policy-options policy-statement agg_route term 1]
user@R0# set from protocol static
user@R0# set from route-filter 1.1.230.0/23 exact
user@R0# set then accept


[edit policy-options policy-statement test_route term 1]
user@R0# set from protocol static
user@R0# set from route-filter 1.1.10.10/32 exact
user@R0# set then accept
```

6. Export the **agg_route** and **test_route** policies from the routing table into BGP.

```
[edit protocols bgp group ext]
user@R0# set export test_route
user@R0# set export agg_route
```

7. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R0# set autonomous-system 65500
```

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

```
user@R0# show interfaces
fe-1/2/0 {
   unit 1 {
      family inet {
         address 1.1.0.1/30;
      }
   }
}
fe-1/2/1 {
   unit 2 {
      family inet {
         address 1.1.1.1/30;
      }
   }
}
```



```
lo0 {
    unit 1 {
        family inet {
            address 10.255.14.179/32;
        }
    }
}
```

```
user@R0# show policy-options
policy-statement agg_route {
    term 1 {
        from {
            protocol static;
            route-filter 1.1.230.0/23 exact;
        }
        then accept;
    }
}
policy-statement test_route {
    term 1 {
        from {
            protocol static;
            route-filter 1.1.10.10/32 exact;
        }
        then accept;
    }
}
```

```
user@R0# show protocols
bgp {
    group ext {
        type external;
        export [ test_route agg_route ];
        peer-as 65000;
        multipath;
        neighbor 1.1.0.2;
        neighbor 1.1.1.2;
    }
}
```

```
user@R0# show routing-options
static {
    route 1.1.10.10/32 reject;
    route 1.1.230.0/23 reject;
}
autonomous-system 65500;
```

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

Configuring Device R1

Step-by-Step Procedure

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

To configure Device R1:

1. Configure the interfaces.

```
[edit interfaces fe-1/2/0 unit 3]
user@R1# set family inet address 1.1.0.2/30

[edit interfaces fe-1/2/1 unit 4]
user@R1# set family inet address 1.12.0.1/30

[edit interfaces fe-1/2/2 unit 5]
user@R1# set family inet address 1.1.1.2/30

[edit interfaces lo0 unit 2]
user@R1# set family inet address 10.255.71.24/32
```

2. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface fe-1/2/1.4
user@R1# set interface 10.255.71.24
```

3. Enable Device R1 to accept the remote next hop.

```
[edit protocols bgp]
user@R1# set accept-remote-nexthop
```

4. Configure IBGP.

```
[edit protocols bgp group int]
user@R1# set type internal
user@R1# set local-address 10.255.71.24
user@R1# set neighbor 10.255.14.177
```

5. Configure EBGP.

```
[edit protocols bgp group ext]
user@R1# set type external
user@R1# set peer-as 65500
user@R1# set neighbor 1.1.0.1
user@R1# set neighbor 1.1.1.1
```

6. Enable multipath BGP between Device R0 and Device R1.

```
[edit protocols bgp group ext]
user@R1# set multipath
```

7. Configure a routing policy that enables a single-hop external BGP peer (Device R1) to accept the remote next-hop 1.1.10.10 for the route to the 1.1.230.0/23 network.

```
[edit policy-options policy-statement agg_route term 1]
user@R1# set from protocol bgp
user@R1# set from route-filter 1.1.230.0/23 exact
```

```
user@R1# set then next-hop 1.1.10.10
user@R1# set then accept
```

8. Import the **agg_route** policy into the routing table on Device R1.

```
[edit protocols bgp group ext]
user@R1# set import agg_route
```

9. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 65000
```

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

```
user@R1# show interfaces
fe-1/2/0 {
  unit 3 {
    family inet {
      address 1.1.0.2/30;
    }
  }
}
fe-1/2/1 {
  unit 4 {
    family inet {
      address 1.12.0.1/30;
    }
  }
}
fe-1/2/2 {
  unit 5 {
    family inet {
      address 1.1.1.2/30;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 10.255.71.24/32;
    }
  }
}
```

```
user@R1# show policy-options
policy-statement agg_route {
  term 1 {
    from {
      protocol bgp;
      route-filter 1.1.230.0/23 exact;
    }
    then {
      next-hop 1.1.10.10;
```

```
        accept;
      }
    }
  }
```

```
user@R1# show protocols
bgp {
  accept-remote-nexthop;
  group ext {
    type external;
    import agg_route;
    peer-as 65500;
    multipath;
    neighbor 1.1.0.1;
    neighbor 1.1.1.1;
  }
  group int {
    type internal;
    local-address 10.255.71.24;
    neighbor 10.255.14.177;
  }
}
ospf {
  area 0.0.0.0 {
    interface fe-1/2/1.4;
    interface 10.255.71.24;
  }
}
```

```
user@R1# show routing-options
autonomous-system 65000;
```

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

Configuring Device R2

Step-by-Step Procedure

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

To configure Device R2:

1. Configure the interfaces.

   ```
   [edit interfaces fe-1/2/0 unit 6]
   user@R2# set family inet address 1.12.0.2/30

   [edit interfaces lo0 unit 3]
   user@R2# set family inet address 10.255.14.177/32
   ```

2. Configure OSPF.

   ```
   [edit protocols ospf area 0.0.0.0]
   user@R2# set interface fe-1/2/0.6
   user@R2# set interface 10.255.14.177
   ```

3. Configure IBGP.

```
[edit protocols bgp group int]
user@R2# set type internal
user@R2# set local-address 10.255.14.177
user@R2# set neighbor 10.255.71.24
```

4. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R1# set autonomous-system 65000
```

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

```
user@R2# show interfaces
fe-1/2/0 {
  unit 6 {
    family inet {
      address 1.12.0.2/30;
    }
  }
}
lo0 {
  unit 3 {
    family inet {
      address 10.255.14.177/32;
    }
  }
}
```

```
user@R2# show protocols
bgp {
  group int {
    type internal;
    local-address 10.255.14.177;
    neighbor 10.255.71.24;
  }
}
ospf {
  area 0.0.0.0 {
    interface fe-1/2/0.6;
    interface 10.255.14.177;
  }
}
```

```
user@R2# show routing-options
autonomous-system 65000;
```

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

Verification

Confirm that the configuration is working properly.

Verifying That the Multipath Route with the Indirect Next Hop Is in the Routing Table

Purpose Verify that Device R1 has a route to the 1.1.230.0/23 network.

Action From operational mode, enter the **show route 1.1.230.0 extensive** command.

```
user@R1> show route 1.1.230.0 extensive
inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
Restart Complete
1.1.230.0/23 (2 entries, 1 announced)
TSI:
KRT in-kernel 1.1.230.0/23 -> {indirect(262142)}
Page 0 idx 1 Type 1 val 9168f6c
    Nexthop: 1.1.10.10
    Localpref: 100
    AS path: [65000] 65500 I
    Communities:
Path 1.1.230.0 from 1.1.0.1 Vector len 4.  Val: 1
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0x90c44d8
                Next-hop reference count: 4
                Source: 1.1.0.1
                Next hop type: Router, Next hop index: 262143
                Next hop: 1.1.0.1 via fe-1/2/0.3, selected
                Next hop: 1.1.1.1 via fe-1/2/2.5
                Protocol next hop: 1.1.10.10
                Indirect next hop: 91c0000 262142
                State: <Active Ext>
                Local AS: 65000 Peer AS: 65500
                Age: 2:55:31    Metric2: 0
                Task: BGP_65500.1.1.0.1+64631
                Announcement bits (3): 2-KRT 3-BGP_RT_Background 4-Resolve tree
1
                AS path: 65500 I
                Accepted Multipath
                Localpref: 100
                Router ID: 10.255.14.179
                Indirect next hops: 1
                        Protocol next hop: 1.1.10.10
                        Indirect next hop: 91c0000 262142
                        Indirect path forwarding next hops: 2
                                Next hop type: Router
                                Next hop: 1.1.0.1 via fe-1/2/0.3
                                Next hop: 1.1.1.1 via fe-1/2/2.5
                        1.1.10.10/32 Originating RIB: inet.0
                          Node path count: 1
                          Forwarding nexthops: 2
                                Nexthop: 1.1.0.1 via fe-1/2/0.3
                                Nexthop: 1.1.1.1 via fe-1/2/2.5
         BGP    Preference: 170/-101
```

```
Next hop type: Indirect
Address: 0x90c44d8
Next-hop reference count: 4
Source: 1.1.1.1
Next hop type: Router, Next hop index: 262143
Next hop: 1.1.0.1 via fe-1/2/0.3, selected
Next hop: 1.1.1.1 via fe-1/2/2.5
Protocol next hop: 1.1.10.10
Indirect next hop: 91c0000 262142
State: <NotBest Ext>
Inactive reason: Not Best in its group - Update source
Local AS: 65000 Peer AS: 65500
Age: 2:55:27    Metric2: 0
Task: BGP_65500.1.1.1.1+53260
AS path: 65500 I
Accepted
Localpref: 100
Router ID: 10.255.14.179
Indirect next hops: 1
        Protocol next hop: 1.1.10.10
        Indirect next hop: 91c0000 262142
        Indirect path forwarding next hops: 2
                Next hop type: Router
                Next hop: 1.1.0.1 via fe-1/2/0.3
                Next hop: 1.1.1.1 via fe-1/2/2.5
        1.1.10.10/32 Originating RIB: inet.0
          Node path count: 1
          Forwarding nexthops: 2
                Nexthop: 1.1.0.1 via fe-1/2/0.3
                Nexthop: 1.1.1.1 via fe-1/2/2.5
```

Meaning The output shows that Device R1 has a route to the 1.1.230.0 network with the multipath feature enabled (**Accepted Multipath**). The output also shows that the route has an indirect next hop of 1.1.10.10.

Deactivating and Reactivating the accept-remote-nexthop Statement

Purpose Make sure that the multipath route with the indirect next hop is removed from the routing table when you deactivate the **accept-remote-nexthop** statement.

Action

1. From configuration mode, enter the **deactivate protocols bgp accept-remote-nexthop** command.

   ```
   user@R1# deactivate protocols bgp accept-remote-nexthop
   user@R1# commit
   ```

2. From operational mode, enter the **show route 1.1.230.0** command.

   ```
   user@R1> show route 1.1.230.0
   ```

3. From configuration mode, reactivate the statement by entering the **activate protocols bgp accept-remote-nexthop** command.

   ```
   user@R1# activate protocols bgp accept-remote-nexthop
   user@R1# commit
   ```

4. From operational mode, reenter the **show route 1.1.230.0** command.

   ```
   user@R1> show route 1.1.230.0

   inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
   ```

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

1.1.230.0/23       *[BGP/170] 03:13:19, localpref 100
                      AS path: 65500 I
                    > to 1.1.0.1 via fe-1/2/0.3
                      to 1.1.1.1 via fe-1/2/2.5
                    [BGP/170] 03:13:15, localpref 100, from 1.1.1.1
                      AS path: 65500 I
                    > to 1.1.0.1 via fe-1/2/0.3
                      to 1.1.1.1 via fe-1/2/2.5
```

Meaning When the **accept-remote-nexthop** statement is deactivated, the multipath route to the 1.1.230.0 network is removed from the routing table .

Related Documentation

- *Example: Overriding the Default BGP Routing Policy on PTX Series Packet Transport Routers*
- *Example: Load Balancing BGP Traffic with Unequal Bandwidth Allocated to the Paths*

Example: Advertising Multiple BGP Paths to a Destination

Understanding the Advertisement of Multiple Paths to a Single Destination in BGP

BGP peers advertise routes to each other in update messages. BGP stores its routes in the Junos OS routing table (**inet.0**). For each prefix in the routing table, the routing protocol process selects a single best path, called the active path. Unless you configure BGP to advertise multiple paths to the same destination, BGP advertises only the active path.

Instead of advertising only the active path to a destination, you can configure BGP to advertise multiple paths to the destination. Within an autonomous system (AS), the availability of multiple exit points to reach a destination provides the following benefits:

- Fault tolerance—Path diversity leads to reduction in restoration time after failure. For instance, a border after receiving multiple paths to the same destination can precompute a backup path and have it ready so that when the primary path becomes invalid, the border routing device can use the backup to quickly restore connectivity. Without a backup path, the restoration time depends on BGP reconvergence, which includes withdraw and advertisement messages in the network before a new best path can be learned.
- Load balancing—The availability of multiple paths to reach the same destination enables load balancing of traffic, if the routing within the AS meets certain constraints.
- Maintenance—The availability of alternate exit points allows for graceful maintenance operation of routers.

The following limitations apply to advertising multiple routes in BGP:

- Address families supported:
  - IPv4 unicast (**family inet unicast**)
  - IPv6 unicast (**family inet6 unicast**)
  - IPv4 labeled unicast (**family inet labeled-unicast**)
  - IPv6 labeled unicast (**family inet6 labeled-unicast**)
- Internal BGP (IBGP) peers only. No support on external BGP (EBGP) peers.
- Master instance only. No support for routing instances.
- Graceful restart supported, but not nonstop active routing (NSR).
- No BGP Monitoring Protocol (BMP) support.
- No support for EBGP sessions between confederations.
- Prefix policies enable you to filter routes on a router that is configured to advertise multiple paths to a destination. Prefix policies can only match prefixes. They cannot match route attributes, and they cannot change the attributes of routes.

Example: Advertising Multiple Paths in BGP

In this example, BGP routers are configured to advertise multiple paths instead of advertising only the active path. Advertising multiple paths in BGP is specified in Internet draft draft-ietf-idr-add-paths-04, *Advertisement of Multiple Paths in BGP*.

Requirements

This example uses the following hardware and software components:

- Eight BGP-enabled devices.
- Five of the BGP-enabled devices do not necessarily need to be routers. For example, they can be EX Series Ethernet Switches.
- Three of the BGP-enabled devices are configured to send multiple paths or receive multiple paths (or both send and receive multiple paths). These three BGP-enabled devices must be M Series Multiservice Edge Routers, MX Series 3D Universal Edge Routers, or T Series Core Routers.
- The three routers must be running Junos OS Release 11.4 or later.

Overview

The following statements are used for configuring multiple paths to a destination:

```
[edit protocols bgp group group-name family family]
add-path {
  receive;
  send {
    path-count number;
    prefix-policy [ policy-names ];
  }
}
```

In this example, Router R5, Router R6, and Router R7 redistribute static routes into BGP. Router R1 and Router R4 are route reflectors. Router R2 and Router R3 are clients to Route Reflector R1. Router R8 is a client to Route Reflector R4.

Route reflection is optional when multiple-path advertisement is enabled in BGP.

With the **add-path send path-count 6** configuration, Router R1 is configured to send up to six paths (per destination) to Router R4.

With the **add-path receive** configuration, Router R4 is configured to receive multiple paths from Router R1.

With the **add-path send path-count 6** configuration, Router R4 is configured to send up to six paths to Router R8.

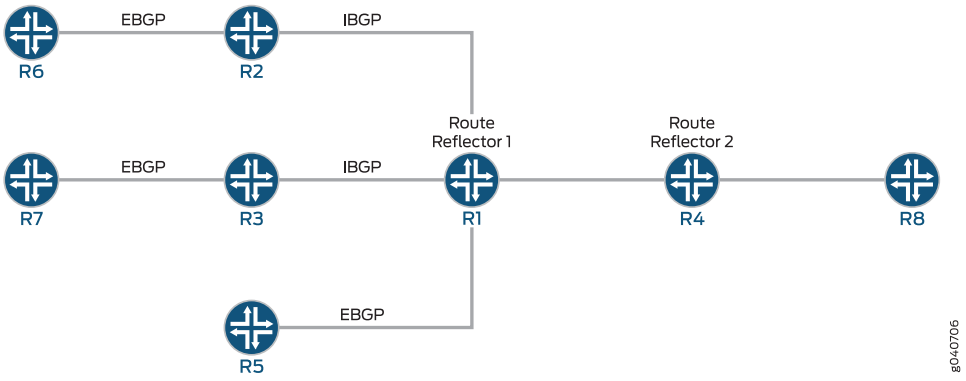
With the **add-path receive** configuration, Router R8 is configured to receive multiple paths from Router R4.

The **add-path send prefix-policy allow_199** policy configuration (along with the corresponding route filter) limits Router R4 to sending multiple paths for only the 199.1.1.1/32 route.

Topology Diagram

Figure 25 on page 250 shows the topology used in this example.

Figure 25: Advertisement of Multiple Paths in BGP



Configuration

CLI Quick Configuration

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

Router R1

```
set interfaces fe-0/0/0 unit 12 family inet address 10.0.12.1/24
set interfaces fe-0/0/1 unit 13 family inet address 10.0.13.1/24
set interfaces fe-1/0/0 unit 14 family inet address 10.0.14.1/24
set interfaces fe-1/2/0 unit 15 family inet address 10.0.15.1/24
set interfaces lo0 unit 10 family inet address 10.0.0.10/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.10
set protocols bgp group rr cluster 10.0.0.10
set protocols bgp group rr neighbor 10.0.0.20
set protocols bgp group rr neighbor 10.0.0.30
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.15.2 local-address 10.0.15.1
set protocols bgp group e1 neighbor 10.0.15.2 peer-as 2
set protocols bgp group rr_rr type internal
set protocols bgp group rr_rr local-address 10.0.0.10
set protocols bgp group rr_rr neighbor 10.0.0.40 family inet unicast add-path send
  path-count 6
set protocols ospf area 0.0.0.0 interface lo0.10 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.12
set protocols ospf area 0.0.0.0 interface fe-0/0/1.13
set protocols ospf area 0.0.0.0 interface fe-1/0/0.14
set protocols ospf area 0.0.0.0 interface fe-1/2/0.15
set routing-options router-id 10.0.0.10
set routing-options autonomous-system 1
```

Router R2

```
set interfaces fe-1/2/0 unit 21 family inet address 10.0.12.2/24
set interfaces fe-1/2/1 unit 26 family inet address 10.0.26.1/24
set interfaces lo0 unit 20 family inet address 10.0.0.20/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.20
set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.26.2 peer-as 2
set protocols ospf area 0.0.0.0 interface lo0.20 passive
```

```
set protocols ospf area 0.0.0.0 interface fe-1/2/0.21
set protocols ospf area 0.0.0.0 interface fe-1/2/1.28
set policy-options policy-statement set_nh_self then next-hop self
set routing-options autonomous-system 1
```

Router R3

```
set interfaces fe-1/0/1 unit 31 family inet address 10.0.13.2/24
set interfaces fe-1/0/2 unit 37 family inet address 10.0.37.1/24
set interfaces lo0 unit 30 family inet address 10.0.0.30/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.30
set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.37.2 peer-as 2
set protocols ospf area 0.0.0.0 interface lo0.30 passive
set protocols ospf area 0.0.0.0 interface fe-1/0/1.31
set protocols ospf area 0.0.0.0 interface fe-1/0/2.37
set policy-options policy-statement set_nh_self then next-hop self
set routing-options autonomous-system 1
```

Router R4

```
set interfaces fe-1/2/0 unit 41 family inet address 10.0.14.2/24
set interfaces fe-1/2/1 unit 48 family inet address 10.0.48.1/24
set interfaces lo0 unit 40 family inet address 10.0.0.40/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.40
set protocols bgp group rr family inet unicast add-path receive
set protocols bgp group rr neighbor 10.0.0.10
set protocols bgp group rr_client type internal
set protocols bgp group rr_client local-address 10.0.0.40
set protocols bgp group rr_client cluster 10.0.0.40
set protocols bgp group rr_client neighbor 10.0.0.80 family inet unicast add-path send
  path-count 6
set protocols bgp group rr_client neighbor 10.0.0.80 family inet unicast add-path send
  prefix-policy allow_199
set protocols ospf area 0.0.0.0 interface fe-1/2/0.41
set protocols ospf area 0.0.0.0 interface lo0.40 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/1.48
set routing-options autonomous-system 1
set policy-options policy-statement allow_199 from route-filter 199.1.1.1/32 exact
set policy-options policy-statement allow_199 then accept
```

Router R5

```
set interfaces fe-1/2/0 unit 51 family inet address 10.0.15.2/24
set interfaces lo0 unit 50 family inet address 10.0.0.50/32
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.15.1 export s2b
set protocols bgp group e1 neighbor 10.0.15.1 peer-as 1
set policy-options policy-statement s2b from protocol static
set policy-options policy-statement s2b from protocol direct
set policy-options policy-statement s2b then as-path-expand 2
set policy-options policy-statement s2b then accept
set routing-options autonomous-system 2
set routing-options static route 199.1.1.1/32 reject
set routing-options static route 198.1.1.1/32 reject
```

Router R6

```
set interfaces fe-1/2/0 unit 62 family inet address 10.0.26.2/24
set interfaces lo0 unit 60 family inet address 10.0.0.60/32
```

```
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.26.1 export s2b
set protocols bgp group e1 neighbor 10.0.26.1 peer-as 1
set policy-options policy-statement s2b from protocol static
set policy-options policy-statement s2b from protocol direct
set policy-options policy-statement s2b then accept
set routing-options autonomous-system 2
set routing-options static route 199.1.1.1/32 reject
set routing-options static route 198.1.1.1/32 reject
```

Router R7

```
set interfaces fe-1/2/0 unit 73 family inet address 10.0.37.2/24
set interfaces lo0 unit 70 family inet address 10.0.0.70/32
set policy-options policy-statement s2b from protocol static
set policy-options policy-statement s2b from protocol direct
set policy-options policy-statement s2b then accept
set protocols bgp group e1 type external
set protocols bgp group e1 neighbor 10.0.37.1 export s2b
set protocols bgp group e1 neighbor 10.0.37.1 peer-as 1
set routing-options autonomous-system 2
set routing-options static route 199.1.1.1/32 reject
```

Router R8

```
set interfaces fe-1/2/0 unit 84 family inet address 10.0.48.2/24
set interfaces lo0 unit 80 family inet address 10.0.0.80/32
set protocols bgp group rr type internal
set protocols bgp group rr local-address 10.0.0.80
set protocols bgp group rr neighbor 10.0.0.40 family inet unicast add-path receive
set protocols ospf area 0.0.0.0 interface lo0.80 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/0.84
set routing-options autonomous-system 1
```

Configuring Router R1

Step-by-Step Procedure

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

To configure Router R1:

1. Configure the interfaces to Router R2, Router R3, Router R4, and Router R5, and configure the loopback (lo0) interface.

   ```
   [edit interfaces]
   user@R1# set fe-0/0/0 unit 12 family inet address 10.0.12.1/24

   user@R1# set fe-0/0/1 unit 13 family inet address 10.0.13.1/24

   user@R1# set fe-1/0/0 unit 14 family inet address 10.0.14.1/24

   user@R1# set fe-1/2/0 unit 15 family inet address 10.0.15.1/24

   user@R1#set lo0 unit 10 family inet address 10.0.0.10/32
   ```

2. Configure BGP on the interfaces, and configure IBGP route reflection.

   ```
   [edit protocols bgp]
   ```

```
user@R1# set group rr type internal
user@R1# set group rr local-address 10.0.0.10
user@R1# set group rr cluster 10.0.0.10
user@R1# set group rr neighbor 10.0.0.20
user@R1# set group rr neighbor 10.0.0.30

user@R1# set group rr_rr type internal
user@R1# set group rr_rr local-address 10.0.0.10

user@R1# set group e1 type external
user@R1# set group e1 neighbor 10.0.15.2 local-address 10.0.15.1
user@R1# set group e1 neighbor 10.0.15.2 peer-as 2
```

3. Configure Router R1 to send up to six paths to its neighbor, Router R4.

   The destination of the paths can be any destination that Router R1 can reach through multiple paths.

```
[edit protocols bgp]
user@R1# set group rr_rr neighbor 10.0.0.40 family inet unicast add-path send
  path-count 6
```

4. Configure OSPF on the interfaces.

```
[edit protocols ospf]
user@R1# set area 0.0.0.0 interface lo0.10 passive
user@R1# set area 0.0.0.0 interface fe-0/0/0.12
user@R1# set area 0.0.0.0 interface fe-0/0/1.13
user@R1# set area 0.0.0.0 interface fe-1/0/0.14
user@R1# set area 0.0.0.0 interface fe-1/2/0.15
```

5. Configure the router ID and the autonomous system number.

```
[edit routing-options]
user@R1# set router-id 10.0.0.10
user@R1# set autonomous-system 1
```

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

```
user@R1# commit
```

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

```
user@R1# show interfaces
fe-0/0/0 {
  unit 12 {
    family inet {
      address 10.0.12.1/24;
    }
  }
}
fe-0/0/1 {
  unit 13 {
```



```
      family inet {
        address 10.0.13.1/24;
      }
    }
  }
  fe-1/0/0 {
    unit 14 {
      family inet {
        address 10.0.14.1/24;
      }
    }
  }
  fe-1/2/0 {
    unit 15 {
      family inet {
        address 10.0.15.1/24;
      }
    }
  }
  lo0 {
    unit 10 {
      family inet {
        address 10.0.0.10/32;
      }
    }
  }
```

```
user@R1# show protocols
bgp {
  group rr {
    type internal;
    local-address 10.0.0.10;
    cluster 10.0.0.10;
    neighbor 10.0.0.20;
    neighbor 10.0.0.30;
  }
  group e1 {
    type external;
    neighbor 10.0.15.2 {
      local-address 10.0.15.1;
      peer-as 2;
    }
  }
  group rr_rr {
    type internal;
    local-address 10.0.0.10;
    neighbor 10.0.0.40 {
      family inet {
        unicast {
          add-path {
            send {
              path-count 6;
            }
          }
        }
      }
```

```
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.10 {
      passive;
    }
    interface fe-0/0/0.12;
    interface fe-0/0/1.13;
    interface fe-1/0/0.14;
    interface fe-1/2/0.15;
  }
}
```

```
user@R1# show routing-options
router-id 10.0.0.10;
autonomous-system 1;
```

Configuring Router R2

Step-by-Step Procedure

To configure Router R2:

1. Configure the loopback (lo0) interface and the interfaces to Router R6 and Router R1.

```
[edit interfaces]
user@R2# set fe-1/2/0 unit 21 family inet address 10.0.12.2/24

user@R2# set fe-1/2/1 unit 26 family inet address 10.0.26.1/24

user@R2# set lo0 unit 20 family inet address 10.0.0.20/32
```

2. Configure BGP and OSPF on Router R2's interfaces.

```
[edit protocols]
user@R2# set bgp group rr type internal
user@R2# set bgp group rr local-address 10.0.0.20

user@R2# set bgp group e1 type external
user@R2# set bgp group e1 neighbor 10.0.26.2 peer-as 2

user@R2# set ospf area 0.0.0.0 interface lo0.20 passive
user@R2# set ospf area 0.0.0.0 interface fe-1/2/0.21
user@R2# set ospf area 0.0.0.0 interface fe-1/2/1.28
```

3. For routes sent from Router R2 to Router R1, advertise Router R2 as the next hop, because Router R1 does not have a route to Router R6's address on the 10.0.26.0/24 network.

```
[edit]
user@R2# set policy-options policy-statement set_nh_self then next-hop self

user@R2# set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
```

4. Configure the autonomous system number.

```
[edit]
user@R2# set routing-options autonomous-system 1
```

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

```
user@R2# commit
```

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

```
user@R2# show interfaces
fe-1/2/0 {
  unit 21 {
    family inet {
      address 10.0.12.2/24;
    }
  }
}
fe-1/2/1 {
  unit 26 {
    family inet {
      address 10.0.26.1/24;
    }
  }
}
lo0 {
  unit 20 {
    family inet {
      address 10.0.0.20/32;
    }
  }
}
```

```
user@R2# show protocols
bgp {
  group rr {
    type internal;
    local-address 10.0.0.20;
    neighbor 10.0.0.10 {
      export set_nh_self;
    }
  }
  group e1 {
    type external;
    neighbor 10.0.26.2 {
      peer-as 2;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.20 {
```



```
        passive;
      }
      interface fe-1/2/0.21;
      interface fe-1/2/1.28;
    }
  }
```

```
user@R2# show policy-options
policy-statement set_nh_self {
  then {
    next-hop self;
  }
}
```

```
user@R2# show routing-options
autonomous-system 1;
```

Configuring Router R3

Step-by-Step Procedure

To configure Router R3:

1. Configure the loopback (lo0) interface and the interfaces to Router R7 and Router R1.

```
[edit interfaces]
user@R3# set fe-1/0/1 unit 31 family inet address 10.0.13.2/24

user@R3# set fe-1/0/2 unit 37 family inet address 10.0.37.1/24

user@R3# set lo0 unit 30 family inet address 10.0.0.30/32
```

2. Configure BGP and OSPF on Router R3's interfaces.

```
[edit protocols]
user@R3# set bgp group rr type internal
user@R3# set bgp group rr local-address 10.0.0.30

user@R3# set bgp group e1 type external
user@R3# set bgp group e1 neighbor 10.0.37.2 peer-as 2

user@R3# set ospf area 0.0.0.0 interface lo0.30 passive
user@R3# set ospf area 0.0.0.0 interface fe-1/0/1.31
user@R3# set ospf area 0.0.0.0 interface fe-1/0/2.37
```

3. For routes sent from Router R3 to Router R1, advertise Router R3 as the next hop, because Router R1 does not have a route to Router R7's address on the 10.0.37.0/24 network.

```
[edit]
user@R3# set policy-options policy-statement set_nh_self then next-hop self

user@R3# set protocols bgp group rr neighbor 10.0.0.10 export set_nh_self
```

4. Configure the autonomous system number.

```
[edit]
user@R3# set routing-options autonomous-system 1
```

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

```
user@R3# commit
```

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

```
user@R3# show interfaces
fe-1/0/1 {
  unit 31 {
    family inet {
      address 10.0.13.2/24;
    }
  }
}
fe-1/0/2 {
  unit 37 {
    family inet {
      address 10.0.37.1/24;
    }
  }
}
lo0 {
  unit 30 {
    family inet {
      address 10.0.0.30/32;
    }
  }
}
```

```
user@R3# show protocols
bgp {
  group rr {
    type internal;
    local-address 10.0.0.30;
    neighbor 10.0.0.10 {
      export set_nh_self;
    }
  }
  group e1 {
    type external;
    neighbor 10.0.37.2 {
      peer-as 2;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.30 {
      passive;
    }
    interface fe-1/0/1.31;
    interface fe-1/0/2.37;
```



```
    }
  }
  user@R3# show policy-options
  policy-statement set_nh_self {
    then {
      next-hop self;
    }
  }

  user@R3# show routing-options
  autonomous-system 1;
```

Configuring Router R4

Step-by-Step Procedure

To configure Router R4:

1. Configure the interfaces to Router R1 and Router R8, and configure the loopback (lo0) interface.

   ```
   [edit interfaces]
   user@R4# set fe-1/2/0 unit 41 family inet address 10.0.14.2/24

   user@R4# set fe-1/2/1 unit 48 family inet address 10.0.48.1/24

   user@R4# set lo0 unit 40 family inet address 10.0.0.40/32
   ```

2. Configure BGP on the interfaces, and configure IBGP route reflection.

   ```
   [edit protocols bgp]
   user@R4# set group rr type internal
   user@R4# set group rr local-address 10.0.0.40
   user@R4# set group rr neighbor 10.0.0.10

   user@R4# set group rr_client type internal
   user@R4# set group rr_client local-address 10.0.0.40
   user@R4# set group rr_client cluster 10.0.0.40
   ```

3. Configure Router R4 to send up to six paths to its neighbor, Router R8.

   The destination of the paths can be any destination that Router R4 can reach through multiple paths.

   ```
   [edit protocols bgp]
   user@R4# set group rr_client neighbor 10.0.0.80 family inet unicast add-path send
     path-count 6
   ```

4. Configure Router R4 to receive multiple paths from its neighbor, Router R1.

   The destination of the paths can be any destination that Router R1 can reach through multiple paths.

   ```
   [edit protocols bgp group rr family inet unicast]
   user@R4# set add-path receive
   ```

5. Configure OSPF on the interfaces.

   ```
   [edit protocols ospf area 0.0.0.0]
   user@R4# set interface fe-1/2/0.41
   ```

```
user@R4# set interface lo0.40 passive
user@R4# set interface fe-1/2/1.48
```

6. Configure a policy that allows Router R4 to send Router R8 multiple paths to the 199.1.1.1/32 route.

   Router R4 receives multiple paths for the 198.1.1.1/32 route and the 199.1.1.1/32 route. However, because of this policy, Router R4 only sends multiple paths for the 199.1.1.1/32 route.

```
[edit protocols bgp group rr_client neighbor 10.0.0.80 family inet unicast]
user@R4# set add-path send prefix-policy allow_199

[edit policy-options policy-statement allow_199]
user@R4# set from route-filter 199.1.1.1/32 exact
user@R4# set then accept
```

7. Configure the autonomous system number.

```
[edit routing-options]
user@R4# set autonomous-system 1
```

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

```
user@R4# commit
```

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

```
user@R4# show interfaces
fe-1/2/0 {
    unit 41 {
        family inet {
            address 10.0.14.2/24;
        }
    }
}
fe-1/2/1 {
    unit 48 {
        family inet {
            address 10.0.48.1/24;
        }
    }
}
lo0 {
    unit 40 {
        family inet {
            address 10.0.0.40/32;
        }
    }
}

user@R4# show protocols
bgp {
    group rr {
```

```
    type internal;
    local-address 10.0.0.40;
    family inet {
      unicast {
        add-path {
          receive;
        }
      }
    }
    neighbor 10.0.0.10;
  }
  group rr_client {
    type internal;
    local-address 10.0.0.40;
    cluster 10.0.0.40;
    neighbor 10.0.0.80 {
      family inet {
        unicast {
          add-path {
            send {
              path-count 6;
              prefix-policy allow_199;
            }
          }
        }
      }
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.40 {
      passive;
    }
    interface fe-1/2/0.41;
    interface fe-1/2/1.48;
  }
}
```

```
user@R4# show policy-options
policy-statement allow_199 {
  from {
    route-filter 199.1.1.1/32 exact;
  }
  then accept;
}
```

```
user@R4# show routing-options
autonomous-system 1;
```

Configuring Router R5

Step-by-Step Procedure

To configure Router R5:

1. Configure the loopback (lo0) interface and the interface to Router R1.

   [edit interfaces]

```
user@R5# set fe-1/2/0 unit 51 family inet address 10.0.15.2/24

user@R5# set lo0 unit 50 family inet address 10.0.0.50/32
```

2. Configure BGP on Router R5's interface.

```
[edit protocols bgp group e1]
user@R5# set type external
user@R5# set neighbor 10.0.15.1 peer-as 1
```

3. Create static routes for redistribution into BGP.

```
[edit routing-options]
user@R5# set static route 199.1.1.1/32 reject
user@R5# set static route 198.1.1.1/32 reject
```

4. Redistribute static and direct routes into BGP.

```
[edit protocols bgp group e1 neighbor 10.0.15.1]
user@R5# set export s2b

[edit policy-options policy-statement s2b]
user@R5# set from protocol static
user@R5# set from protocol direct
user@R5# set then as-path-expand 2
user@R5# set then accept
```

5. Configure the autonomous system number.

```
[edit routing-options]
user@R5# set autonomous-system 2
```

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

```
user@R5# commit
```

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

```
user@R5# show interfaces
fe-1/2/0 {
    unit 51 {
        family inet {
            address 10.0.15.2/24;
        }
    }
}
lo0 {
    unit 50 {
        family inet {
            address 10.0.0.50/32;
        }
    }
}
```

```
user@R5# show protocols
bgp {
  group e1 {
    type external;
    neighbor 10.0.15.1 {
      export s2b;
      peer-as 1;
    }
  }
}
```

```
user@R5# show policy-options
policy-statement s2b {
  from protocol [ static direct ];
  then {
    as-path-expand 2;
    accept;
  }
}
```

```
user@R5# show routing-options
static {
  route 198.1.1.1/32 reject;
  route 199.1.1.1/32 reject;
}
autonomous-system 2;
```

Configuring Router R6

Step-by-Step Procedure

To configure Router R6:

1. Configure the loopback (lo0) interface and the interface to Router R2.

```
[edit interfaces]
user@R6# set fe-1/2/0 unit 62 family inet address 10.0.26.2/24

user@R6# set lo0 unit 60 family inet address 10.0.0.60/32
```

2. Configure BGP on Router R6's interface.

```
[edit protocols]
user@R6# set bgp group e1 type external
user@R6# set bgp group e1 neighbor 10.0.26.1 peer-as 1
```

3. Create static routes for redistribution into BGP.

```
[edit]
user@R6# set routing-options static route 199.1.1.1/32 reject
user@R6# set routing-options static route 198.1.1.1/32 reject
```

4. Redistribute static and direct routes from Router R6's routing table into BGP.

```
[edit protocols bgp group e1 neighbor 10.0.26.1]
user@R6# set export s2b

[edit policy-options policy-statement s2b]
user@R6# set from protocol static
user@R6# set from protocol direct
```

```
user@R6# set then accept
```

5. Configure the autonomous system number.

```
[edit routing-options]
user@R6# set autonomous-system 2
```

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

```
user@R6# commit
```

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

```
user@R6# show interfaces
fe-1/2/0 {
  unit 62 {
    family inet {
      address 10.0.26.2/24;
    }
  }
}
lo0 {
  unit 60 {
    family inet {
      address 10.0.0.60/32;
    }
  }
}
```

```
user@R6# show protocols
bgp {
  group e1 {
    type external;
    neighbor 10.0.26.1 {
      export s2b;
      peer-as 1;
    }
  }
}
```

```
user@R6# show policy-options
policy-statement s2b {
  from protocol [ static direct ];
  then accept;
}
```

```
user@R6# show routing-options
static {
  route 198.1.1.1/32 reject;
  route 199.1.1.1/32 reject;
}
autonomous-system 2;
```


Configuring Router R7

Step-by-Step Procedure

To configure Router R7:

1. Configure the loopback (lo0) interface and the interface to Router R3.

```
[edit interfaces]
user@R7# set fe-1/2/0 unit 73 family inet address 10.0.37.2/24

user@R7# set lo0 unit 70 family inet address 10.0.0.70/32
```

2. Configure BGP on Router R7's interface.

```
[edit protocols bgp group e1]
user@R7# set type external
user@R7# set neighbor 10.0.37.1 peer-as 1
```

3. Create a static route for redistribution into BGP.

```
[edit]
user@R7# set routing-options static route 199.1.1.1/32 reject
```

4. Redistribute static and direct routes from Router R7's routing table into BGP.

```
[edit protocols bgp group e1 neighbor 10.0.37.1]
user@R7# set export s2b

[edit policy-options policy-statement s2b]
user@R7# set from protocol static
user@R7# set from protocol direct
user@R7# set then accept
```

5. Configure the autonomous system number.

```
[edit routing-options]
user@R7# set autonomous-system 2
```

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

```
user@R7# commit
```

Results

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

```
user@R7# show interfaces
fe-1/2/0 {
    unit 73 {
        family inet {
            address 10.0.37.2/24;
        }
    }
}
lo0 {
    unit 70 {
        family inet {
```

```
        address 10.0.0.70/32;
      }
    }
  }
```

```
user@R7# show protocols
bgp {
  group e1 {
    type external;
    neighbor 10.0.37.1 {
      export s2b;
      peer-as 1;
    }
  }
}
```

```
user@R7# show policy-options
policy-statement s2b {
  from protocol [ static direct ];
  then accept;
}
```

```
user@R7# show routing-options
static {
  route 199.1.1.1/32 reject;
}
autonomous-system 2;
```

Configuring Router R8

Step-by-Step Procedure

To configure Router R8:

1. Configure the loopback (lo0) interface and the interface to Router R4.

```
[edit interfaces]
user@R8# set fe-1/2/0 unit 84 family inet address 10.0.48.2/24


user@R8# set lo0 unit 80 family inet address 10.0.0.80/32
```

2. Configure BGP and OSPF on Router R8's interface.

```
[edit protocols]
user@R8# set bgp group rr type internal
user@R8# set bgp group rr local-address 10.0.0.80


user@R8# set ospf area 0.0.0.0 interface lo0.80 passive
user@R8# set ospf area 0.0.0.0 interface fe-1/2/0.84
```

3. Configure Router R8 to receive multiple paths from its neighbor, Router R4.

   The destination of the paths can be any destination that Router R4 can reach through multiple paths.

```
[edit protocols]
user@R8# set bgp group rr neighbor 10.0.0.40 family inet unicast add-path receive
```

4. Configure the autonomous system number.

```
[edit]
```

```
user@R8# set routing-options autonomous-system 1
```

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

```
user@R8# commit
```

Results

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

```
user@R8# show interfaces
fe-1/2/0 {
    unit 84 {
        family inet {
            address 10.0.48.2/24;
        }
    }
}
lo0 {
    unit 80 {
        family inet {
            address 10.0.0.80/32;
        }
    }
}
```

```
user@R8# show protocols
bgp {
    group rr {
        type internal;
        local-address 10.0.0.80;
        neighbor 10.0.0.40 {
            family inet {
                unicast {
                    add-path {
                        receive;
                    }
                }
            }
        }
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.80 {
            passive;
        }
        interface fe-1/2/0.84;
    }
}
```

```
user@R8# show routing-options
autonomous-system 1;
```

Verification

Confirm that the configuration is working properly.

Verifying That the BGP Peers Have the Ability to Send and Receive Multiple Paths

Purpose Make sure that one or both of the following strings appear in the output of the **show bgp neighbor** command:

- **NLRI's for which peer can receive multiple paths: inet-unicast**
- **NLRI's for which peer can send multiple paths: inet-unicast**

Action

```
user@R1> show bgp neighbor 10.0.0.40
Peer: 10.0.0.40+179 AS 1       Local: 10.0.0.10+65237 AS 1
  Type: Internal    State: Established    Flags: <Sync>
...  NLRI's for which peer can receive multiple paths: inet-unicast
...

user@R4> show bgp neighbor 10.0.0.10
Peer: 10.0.0.10+65237 AS 1     Local: 10.0.0.40+179 AS 1
  Type: Internal    State: Established    Flags: <Sync>
...
  NLRI's for which peer can send multiple paths: inet-unicast
...

user@R4> show bgp neighbor 10.0.0.80
Peer: 10.0.0.80+55416 AS 1     Local: 10.0.0.40+179 AS 1
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  ,,,
  NLRI's for which peer can receive multiple paths: inet-unicast
  ...

user@R8> show bgp neighbor 10.0.0.40
Peer: 10.0.0.40+179 AS 1       Local: 10.0.0.80+55416 AS 1
  Type: Internal    State: Established    Flags: <Sync>
  ...
  NLRI's for which peer can send multiple paths: inet-unicast
  ...
```

Verifying That Router R1 Is Advertising Multiple Paths

Purpose Make sure that multiple paths to the 198.1.1.1/32 destination and multiple paths to the 199.1.1.1/32 destination are advertised to Router R4.

Action

```
user@R1> show route advertising-protocol bgp 10.0.0.40
inet.0: 21 destinations, 25 routes (21 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.50/32            10.0.15.2                    100        2 2 I
* 10.0.0.60/32            10.0.0.20                    100        2 I
* 10.0.0.70/32            10.0.0.30                    100        2 I
* 198.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.15.2                    100        2 2 I
* 199.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.0.30                    100        2 I
                          10.0.15.2                    100        2 2 I
* 200.1.1.0/30            10.0.0.20                    100        2 I
```

Meaning When you see one prefix and more than one next hop, it means that multiple paths are advertised to Router R4.

Verifying That Router R4 Is Receiving and Advertising Multiple Paths

Purpose Make sure that multiple paths to the 199.1.1.1/32 destination are received from Router R1 and advertised to Router R8. Make sure that multiple paths to the 198.1.1.1/32 destination are received from Router R1, but only one path to this destination is advertised to Router R8.

Action

```
user@R4> show route receive-protocol bgp 10.0.0.10
inet.0: 19 destinations, 22 routes (19 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.50/32            10.0.15.2                    100        2 2 I
* 10.0.0.60/32            10.0.0.20                    100        2 I
* 10.0.0.70/32            10.0.0.30                    100        2 I
* 198.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.15.2                    100        2 2 I
* 199.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.0.30                    100        2 I
                          10.0.15.2                    100        2 2 I
* 200.1.1.0/30            10.0.0.20                    100        2 I
```

```
user@R4> show route advertising-protocol bgp 10.0.0.80
inet.0: 19 destinations, 22 routes (19 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.50/32            10.0.15.2                    100        2 2 I
* 10.0.0.60/32            10.0.0.20                    100        2 I
* 10.0.0.70/32            10.0.0.30                    100        2 I
* 198.1.1.1/32            10.0.0.20                    100        2 I
* 199.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.0.30                    100        2 I
                          10.0.15.2                    100        2 2 I
* 200.1.1.0/30            10.0.0.20                    100        2 I
```

Meaning The **show route receive-protocol** command shows that Router R4 receives two paths to the 198.1.1.1/32 destination and three paths to the 199.1.1.1/32 destination. The **show route advertising-protocol** command shows that Router R4 advertises only one path to the 198.1.1.1/32 destination and advertises all three paths to the 199.1.1.1/32 destination.

Because of the prefix policy that is applied to Router R4, Router R4 does not advertise multiple paths to the 198.1.1.1/32 destination. Router R4 advertises only one path to the 198.1.1.1/32 destination even though it receives multiple paths to this destination.

Verifying That Router R8 Is Receiving Multiple Paths

Purpose Make sure that Router R8 receives multiple paths to the 199.1.1.1/32 destination through Router R4. Make sure that Router R8 receives only one path to the 198.1.1.1/32 destination through Router R4.

Action

```
user@R8> show route receive-protocol bgp 10.0.0.40
inet.0: 18 destinations, 20 routes (18 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.0.50/32            10.0.15.2                    100        2 2 I
* 10.0.0.60/32            10.0.0.20                    100        2 I
* 10.0.0.70/32            10.0.0.30                    100        2 I
* 198.1.1.1/32            10.0.0.20                    100        2 I
* 199.1.1.1/32            10.0.0.20                    100        2 I
                          10.0.0.30                    100        2 I
                          10.0.15.2                    100        2 2 I
* 200.1.1.0/30            10.0.0.20                    100        2 I
```

Checking the Path ID

Purpose On the downstream devices, Router R4 and Router R8, verify that a path ID uniquely identifies the path. Look for the **Addpath Path ID:** string.

Action `user@R4>` **show route 199.1.1.1/32 detail**

```
inet.0: 18 destinations, 20 routes (18 active, 0 holddown, 0 hidden)
199.1.1.1/32 (3 entries, 3 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 9
                Source: 10.0.0.10
                Next hop type: Router, Next hop index: 676
                Next hop: 10.0.14.1 via lt-1/2/0.41, selected
                Protocol next hop: 10.0.0.20
                Indirect next hop: 92041c8 262146
                State: <Active Int Ext>
                Local AS:     1 Peer AS:     1
                Age: 1:44:37    Metric2: 2
                Task: BGP_1.10.0.0.10+65237
                Announcement bits (3): 2-KRT 3-BGP RT Background 4-Resolve tree
1
                AS path: 2 I (Originator) Cluster list:  10.0.0.10
                AS path:  Originator ID: 10.0.0.20
                Accepted
                Localpref: 100
                Router ID: 10.0.0.10
                Addpath Path ID: 1
         BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 4
                Source: 10.0.0.10
                Next hop type: Router, Next hop index: 676
                Next hop: 10.0.14.1 via lt-1/2/0.41, selected
                Protocol next hop: 10.0.0.30
                Indirect next hop: 92042ac 262151
                State: <NotBest Int Ext>
                Inactive reason: Not Best in its group - Router ID
                Local AS:     1 Peer AS:     1
                Age: 1:44:37    Metric2: 2
                Task: BGP_1.10.0.0.10+65237
                Announcement bits (1): 3-BGP RT Background
                AS path: 2 I (Originator) Cluster list:  10.0.0.10
                AS path:  Originator ID: 10.0.0.30
                Accepted
                Localpref: 100
                Router ID: 10.0.0.10
                Addpath Path ID: 2
         BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 4
                Source: 10.0.0.10
                Next hop type: Router, Next hop index: 676
                Next hop: 10.0.14.1 via lt-1/2/0.41, selected
                Protocol next hop: 10.0.15.2
                Indirect next hop: 92040e4 262150
                State: <Int Ext>
                Inactive reason: AS path
                Local AS:     1 Peer AS:     1
                Age: 1:44:37    Metric2: 2
                Task: BGP_1.10.0.0.10+65237
                Announcement bits (1): 3-BGP RT Background
                AS path: 2 2 I
                Accepted
```

```
                Localpref: 100
                Router ID: 10.0.0.10
                Addpath Path ID: 3

user@R8> show route 199.1.1.1/32 detail

inet.0: 17 destinations, 19 routes (17 active, 0 holddown, 0 hidden)
199.1.1.1/32 (3 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 9
                Source: 10.0.0.40
                Next hop type: Router, Next hop index: 1045
                Next hop: 10.0.48.1 via lt-1/2/0.84, selected
                Protocol next hop: 10.0.0.20
                Indirect next hop: 91fc0e4 262148
                State: <Active Int Ext>
                Local AS:     1 Peer AS:     1
                Age: 1:56:51    Metric2: 3
                Task: BGP_1.10.0.0.40+179
                Announcement bits (2): 2-KRT 4-Resolve tree 1
                AS path: 2 I (Originator) Cluster list:  10.0.0.40 10.0.0.10
                AS path:  Originator ID: 10.0.0.20
                Accepted
                Localpref: 100
                Router ID: 10.0.0.40
                Addpath Path ID: 1
         BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 4
                Source: 10.0.0.40
                Next hop type: Router, Next hop index: 1045
                Next hop: 10.0.48.1 via lt-1/2/0.84, selected
                Protocol next hop: 10.0.0.30
                Indirect next hop: 91fc1c8 262152
                State: <NotBest Int Ext>
                Inactive reason: Not Best in its group - Router ID
                Local AS:     1 Peer AS:     1
                Age: 1:56:51    Metric2: 3
                Task: BGP_1.10.0.0.40+179
                AS path: 2 I (Originator) Cluster list:  10.0.0.40 10.0.0.10
                AS path:  Originator ID: 10.0.0.30
                Accepted
                Localpref: 100
                Router ID: 10.0.0.40
                Addpath Path ID: 2
         BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 4
                Source: 10.0.0.40
                Next hop type: Router, Next hop index: 1045
                Next hop: 10.0.48.1 via lt-1/2/0.84, selected
                Protocol next hop: 10.0.15.2
                Indirect next hop: 91fc2ac 262153
                State: <Int Ext>
                Inactive reason: AS path
                Local AS:     1 Peer AS:     1
                Age: 1:56:51    Metric2: 3
                Task: BGP_1.10.0.0.40+179
                AS path: 2 2 I (Originator) Cluster list:  10.0.0.40
                AS path:  Originator ID: 10.0.0.10
```

```
Accepted
Localpref: 100
Router ID: 10.0.0.40
Addpath Path ID: 3
```

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

CHAPTER 7

IBGP Scaling Configuration

Example: Configuring BGP Route Reflectors

Understanding BGP Route Reflectors

Because of the internal BGP (IBGP) full-mesh requirement, most networks use route reflectors to simplify configuration. The formula to compute the number of sessions required for a full mesh is v * (v - 1)/2, where v is the number of BGP-enabled devices. The full-mesh model does not scale well. Using a route reflector, you group routers into clusters, which are identified by numeric identifiers unique to the autonomous system (AS). Within the cluster, you must configure a BGP session from a single router (the route reflector) to each internal peer. With this configuration, the IBGP full-mesh requirement is met.

To use route reflection in an AS, you designate one or more routers as a route reflector—typically, one per point of presence (POP). Route reflectors have the special BGP ability to readvertise routes learned from an internal peer to other internal peers. So rather than requiring all internal peers to be fully meshed with each other, route reflection requires only that the route reflector be fully meshed with all internal peers. The route reflector and all of its internal peers form a cluster, as shown in Figure 26 on page 276.

NOTE: For some Juniper Networks devices, you must have an Advanced BGP Feature license installed on each device that uses a route reflector. For license details, see the *Junos OS Initial Configuration Guide for Security Devices*.

Figure 26: Simple Route Reflector Topology (One Cluster)

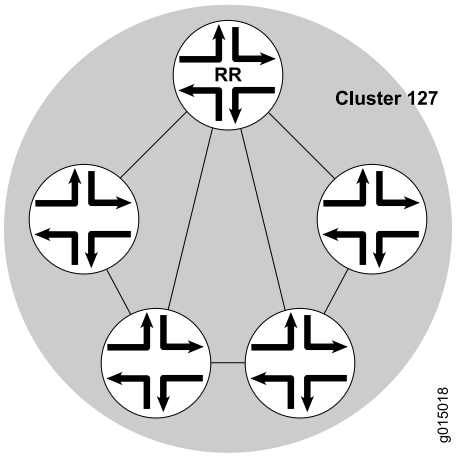


Figure 26 on page 276 shows Router RR configured as the route reflector for Cluster 127. The other routers are designated internal peers within the cluster. BGP routes are advertised to Router RR by any of the internal peers. RR then readvertises those routes to all other peers within the cluster.

You can configure multiple clusters and link them by configuring a full mesh of route reflectors (see Figure 27 on page 276).

Figure 27: Basic Route Reflection (Multiple Clusters)

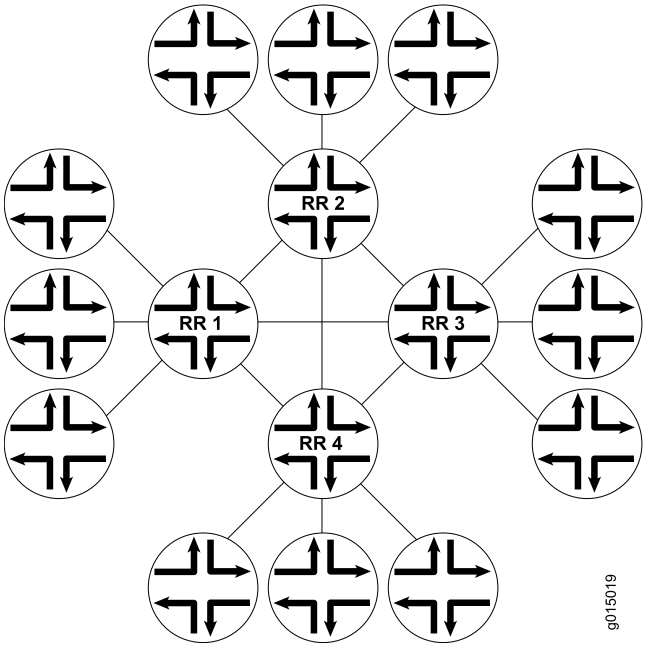


Figure 27 on page 276 shows Route Reflectors RR 1, RR 2, RR 3, and RR 4 as fully meshed internal peers. When a router advertises a route to RR 1, RR 1 readvertises the route to the other route reflectors, which, in turn, readvertise the route to the remaining routers within the AS. Route reflection allows the route to be propagated throughout the AS without the scaling problems created by the full mesh requirement.

However, as clusters become large, a full mesh with a route reflector becomes difficult to scale, as does a full mesh between route reflectors. To help offset this problem, you can group clusters of routers together into clusters of clusters for hierarchical route reflection (see Figure 28 on page 277).

Figure 28: Hierarchical Route Reflection (Clusters of Clusters)

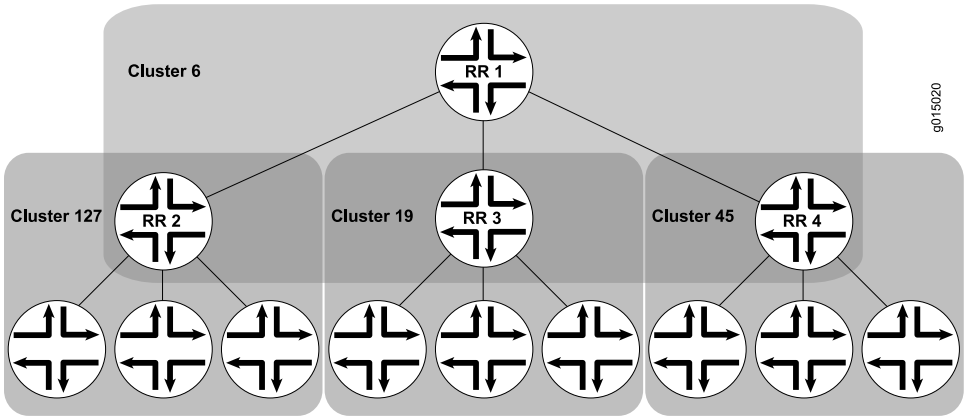


Figure 28 on page 277 shows RR 2, RR 3, and RR 4 as the route reflectors for Clusters 127, 19, and 45, respectively. Rather than fully mesh those route reflectors, the network administrator has configured them as part of another cluster (Cluster 6) for which RR 1 is the route reflector. When a router advertises a route to RR 2, RR 2 readvertises the route to all the routers within its own cluster, and then readvertises the route to RR 1. RR 1 readvertises the route to the routers in its cluster, and those routers propagate the route down through their clusters.

Example: Configuring a Route Reflector

This example shows how to configure a route reflector.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

Generally, internal BGP (IBGP)-enabled devices need to be fully meshed, because IBGP does not readvertise updates to other IBGP-enabled devices. The full mesh is a logical mesh achieved through configuration of multiple **neighbor** statements on each IBGP-enabled device. The full mesh is not necessarily a physical full mesh. Maintaining a full mesh (logical or physical) does not scale well in large deployments.

Figure 29 on page 279 shows an IBGP network with Device A acting as a route reflector. Device B and Device C are clients of the route reflector. Device D and Device E are outside the cluster, so they are nonclients of the route reflector.

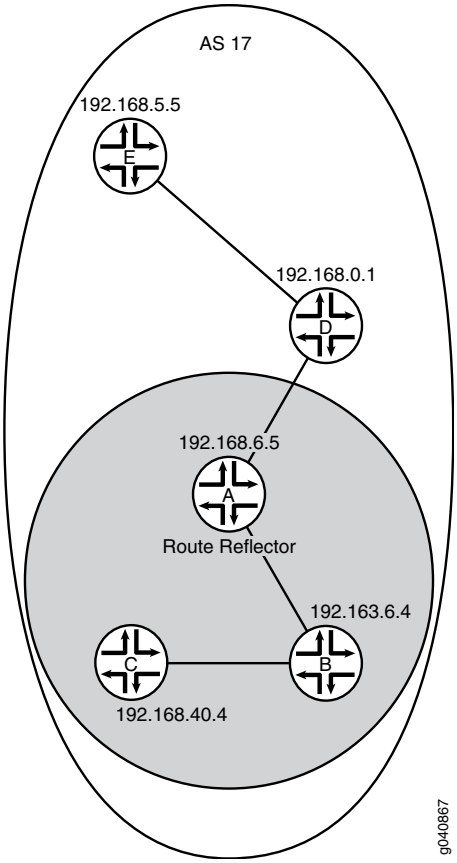
On Device A (the route reflector), you must form peer relationships with all of the IBGP-enabled devices by including the **neighbor** statement for the clients (Device B and Device C) and the nonclients (Device D and Device E). You must also include the **cluster** statement and a cluster identifier. The cluster identifier can be any 32-bit value. This example uses the loopback interface IP address of the route reflector.

On Device B and Device C, the route reflector clients, you only need one **neighbor** statement that forms a peer relationship with the route reflector, Device A.

On Device D and Device E, the nonclients, you need a **neighbor** statement for each nonclient device (D-to-E and E-to-D). You also need a **neighbor** statement for the route reflector (D-to-A and E-to-A). Device D and Device E do not need **neighbor** statements for the client devices (Device B and Device C).

TIP: Device D and Device E are considered to be nonclients because they have explicitly configured peer relationships with each other. To make them RRroute reflector clients, remove the neighbor 192.168.5.5 statement from the configuration on Device D, and remove the neighbor 192.168.0.1 statement from the configuration on Device E.

Figure 29: IBGP Network Using a Route Reflector



Configuration

CLI Quick Configuration

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

Device A

```
set interfaces fe-0/0/0 unit 1 description to-B
set interfaces fe-0/0/0 unit 1 family inet address 10.10.10.1/30
set interfaces fe-0/0/1 unit 3 description to-D
set interfaces fe-0/0/1 unit 3 family inet address 10.10.10.9/30
set interfaces lo0 unit 1 family inet address 192.168.6.5/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.6.5
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers cluster 192.168.6.5
set protocols bgp group internal-peers neighbor 192.163.6.4
set protocols bgp group internal-peers neighbor 192.168.40.4
set protocols bgp group internal-peers neighbor 192.168.0.1
set protocols bgp group internal-peers neighbor 192.168.5.5
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.1
set protocols ospf area 0.0.0.0 interface fe-0/0/1.3
set policy-options policy-statement send-ospf term 2 from protocol ospf
```

```
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.6.5
set routing-options autonomous-system 17
```

Device B

```
set interfaces fe-0/0/0 unit 2 description to-A
set interfaces fe-0/0/0 unit 2 family inet address 10.10.10.2/30
set interfaces fe-0/0/1 unit 5 description to-C
set interfaces fe-0/0/1 unit 5 family inet address 10.10.10.5/30
set interfaces lo0 unit 2 family inet address 192.163.6.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.163.6.4
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.2
set protocols ospf area 0.0.0.0 interface fe-0/0/1.5
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.163.6.4
set routing-options autonomous-system 17
```

Device C

```
set interfaces fe-0/0/0 unit 6 description to-B
set interfaces fe-0/0/0 unit 6 family inet address 10.10.10.6/30
set interfaces lo0 unit 3 family inet address 192.168.40.4/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.40.4
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.6
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.40.4
set routing-options autonomous-system 17
```

Device D

```
set interfaces fe-0/0/0 unit 4 description to-A
set interfaces fe-0/0/0 unit 4 family inet address 10.10.10.10/30
set interfaces fe-0/0/1 unit 7 description to-E
set interfaces fe-0/0/1 unit 7 family inet address 10.10.10.13/30
set interfaces lo0 unit 4 family inet address 192.168.0.1/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.0.1
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols bgp group internal-peers neighbor 192.168.5.5
set protocols ospf area 0.0.0.0 interface lo0.4 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.4
set protocols ospf area 0.0.0.0 interface fe-0/0/1.7
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 17
```

Device E

```
set interfaces fe-0/0/0 unit 8 description to-D
set interfaces fe-0/0/0 unit 8 family inet address 10.10.10.14/30
```

```
set interfaces lo0 unit 5 family inet address 192.168.5.5/32
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 192.168.5.5
set protocols bgp group internal-peers export send-ospf
set protocols bgp group internal-peers neighbor 192.168.0.1
set protocols bgp group internal-peers neighbor 192.168.6.5
set protocols ospf area 0.0.0.0 interface lo0.5 passive
set protocols ospf area 0.0.0.0 interface fe-0/0/0.8
set policy-options policy-statement send-ospf term 2 from protocol ospf
set policy-options policy-statement send-ospf term 2 then accept
set routing-options router-id 192.168.5.5
set routing-options autonomous-system 17
```

Configuring the Route Reflector

Step-by-Step Procedure

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

To configure IBGP in the network using Juniper Networks Device A as a route reflector:

1. Configure the interfaces.

```
[edit interfaces]
user@A# set fe-0/0/0 unit 1 description to-B
user@A# set fe-0/0/0 unit 1 family inet address 10.10.10.1/30
user@A# set fe-0/0/1 unit 3 description to-D
user@A# set fe-0/0/1 unit 3 family inet address 10.10.10.9/30
user@A# set lo0 unit 1 family inet address 192.168.6.5/32
```

2. Configure BGP, including the cluster identifier and neighbor relationships with all IBGP-enabled devices in the autonomous system (AS).

   Also apply the policy that redistributes OSPF routes into BGP.

```
[edit protocols bgp group internal-peers]
user@A# set type internal
user@A# set local-address 192.168.6.5
user@A# set export send-ospf
user@A# set cluster 192.168.6.5
user@A# set neighbor192.163.6.4
user@A# set neighbor 192.168.40.4
user@A# set neighbor 192.168.0.1
user@A# set neighbor 192.168.5.5
```

3. Configure static routing or an interior gateway protocol (IGP).

   This example uses OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@A# set interface lo0.1 passive
user@A# set interface fe-0/0/0.1
user@A# set interface fe-0/0/1.3
```

4. Configure the policy that redistributes OSPF routes into BGP.

```
[edit policy-options policy-statement send-ospf term 2]
user@A# set from protocol ospf
user@A# set then accept
```

5. Configure the router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@A# set router-id 192.168.6.5
user@A# set autonomous-system 17
```

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

```
user@A# show interfaces
fe-0/0/0 {
  unit 1 {
    description to-B;
    family inet {
      address 10.10.10.1/30;
    }
  }
}
fe-0/0/1 {
  unit 3 {
    description to-D;
    family inet {
      address 10.10.10.9/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 192.168.6.5/32;
    }
  }
}
```

```
user@A# show protocols
bgp {
  group internal-peers {
    type internal;
    local-address 192.168.6.5;
    export send-ospf;
    cluster 192.168.6.5;
    neighbor 192.163.6.4;
    neighbor 192.168.40.4;
    neighbor 192.168.0.1;
    neighbor 192.168.5.5;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.1 {
      passive;
    }
    interface fe-0/0/0.1;
    interface fe-0/0/1.3;
```



```
  }
}
```

```
user@A# show policy-options
policy-statement send-ospf {
  term 2 {
    from protocol ospf;
    then accept;
  }
}
```

```
user@A# show routing-options
router-id 192.168.6.5;
autonomous-system 17;
```

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

NOTE: Repeat these steps for each nonclient BGP peer within the cluster that you are configuring, if the other nonclient devices are from Juniper Networks. Otherwise, consult the device's documentation for instructions.

Configuring Client Peers

Step-by-Step Procedure

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

To configure client peers:

1. Configure the interfaces.

   ```
   [edit interfaces]
   user@B# set fe-0/0/0 unit 2 description to-A
   user@B# set fe-0/0/0 unit 2 family inet address 10.10.10.2/30
   user@B# set fe-0/0/1 unit 5 description to-C
   user@B# set fe-0/0/1 unit 5 family inet address 10.10.10.5/30
   user@B# set lo0 unit 2 family inet address 192.163.6.4/32
   ```

2. Configure the BGP neighbor relationship with the route reflector.

   Also apply the policy that redistributes OSPF routes into BGP.

   ```
   [edit protocols bgp group internal-peers]
   user@B# set type internal
   user@B# set local-address 192.163.6.4
   user@B# set export send-ospf
   user@B# set neighbor 192.168.6.5
   ```

3. Configure OSPF.

   ```
   [edit protocols ospf area 0.0.0.0]
   user@B# set interface lo0.2 passive
   user@B# set interface fe-0/0/0.2
   user@B# set interface fe-0/0/1.5
   ```

4. Configure the policy that redistributes OSPF routes into BGP.

```
[edit policy-options policy-statement send-ospf term 2]
user@B# set from protocol ospf
user@B# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@B# set router-id 192.163.6.4
user@B# set autonomous-system 17
```

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

```
user@B# show interfaces
fe-0/0/0 {
  unit 2 {
    description to-A;
    family inet {
      address 10.10.10.2/30;
    }
  }
}
fe-0/0/1 {
  unit 5 {
    description to-C;
    family inet {
      address 10.10.10.5/30;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 192.163.6.4/32;
    }
  }
}
```

```
user@B# show protocols
bgp {
  group internal-peers {
    type internal;
    local-address 192.163.6.4;
    export send-ospf;
    neighbor 192.168.6.5;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.2 {
      passive;
    }
    interface fe-0/0/0.2;
    interface fe-0/0/1.5;
```



```
    }
}
```

```
user@B# show policy-options
policy-statement send-ospf {
    term 2 {
        from protocol ospf;
        then accept;
    }
}
```

```
user@B# show routing-options
router-id 192.163.6.4;
autonomous-system 17;
```

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

NOTE: Repeat these steps for each client BGP peer within the cluster that you are configuring if the other client devices are from Juniper Networks. Otherwise, consult the device's documentation for instructions.

Configuring Nonclient Peers

Step-by-Step Procedure

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

To configure nonclient peers:

1. Configure the interfaces.

   ```
   [edit interfaces]
   user@D# set fe-0/0/0 unit 4 description to-A
   user@D# set fe-0/0/0 unit 4 family inet address 10.10.10.10/30
   user@D# set fe-0/0/1 unit 7 description to-E
   user@D# set fe-0/0/1 unit 7 family inet address 10.10.10.13/30
   user@D# set lo0 unit 4 family inet address 192.168.0.1/32
   ```

2. Configure the BGP neighbor relationships with the RRroute reflector and with the other nonclient peers.

   Also apply the policy that redistributes OSPF routes into BGP.

   ```
   [edit protocols bgp group internal-peers]
   user@D# set type internal
   user@D# set local-address 192.168.0.1
   user@D# set export send-ospf
   user@D# set neighbor 192.168.6.5
   user@D# set neighbor 192.168.5.5
   ```

3. Configure OSPF.

   ```
   [edit protocols ospf area 0.0.0.0]
   user@D# set interface lo0.4 passive
   user@D# set interface fe-0/0/0.4
   ```

```
user@D# set interface fe-0/0/1.7
```

4. Configure the policy that redistributes OSPF routes into BGP.

```
[edit policy-options policy-statement send-ospf term 2]
user@D# set from protocol ospf
user@D# set then accept
```

5. Configure the router ID and the AS number.

```
[edit routing-options]
user@D# set router-id 192.168.0.1
user@D# set autonomous-system 17
```

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

```
user@D# show interfaces
fe-0/0/0 {
  unit 4 {
    description to-A;
    family inet {
      address 10.10.10.10/30;
    }
  }
}
fe-0/0/1 {
  unit 7 {
    description to-E;
    family inet {
      address 10.10.10.13/30;
    }
  }
}
lo0 {
  unit 4 {
    family inet {
      address 192.168.0.1/32;
    }
  }
}
```

```
user@D# show protocols
bgp {
  group internal-peers {
    type internal;
    local-address 192.168.0.1;
    export send-ospf;
    neighbor 192.168.6.5;
    neighbor 192.168.5.5;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.4 {
```



```
        passive;
      }
      interface fe-0/0/0.4;
      interface fe-0/0/1.7;
    }
  }
```

```
user@D# show policy-options
policy-statement send-ospf {
  term 2 {
    from protocol ospf;
    then accept;
  }
}
```

```
user@D# show routing-options
router-id 192.168.0.1;
autonomous-system 17;
```

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

NOTE: Repeat these steps for each nonclient BGP peer within the cluster that you are configuring if the other nonclient devices are from Juniper Networks. Otherwise, consult the device's documentation for instructions.

Verification

Confirm that the configuration is working properly.

- Verifying BGP Neighbors on page 287
- Verifying BGP Groups on page 290
- Verifying BGP Summary Information on page 290
- Verifying Routing Table Information on page 290

Verifying BGP Neighbors

Purpose Verify that BGP is running on configured interfaces and that the BGP session is established for each neighbor address.

Action From operational mode, enter the **show bgp neighbor** command.

```
user@A> show bgp neighbor
Peer: 192.163.6.4+179 AS 17    Local: 192.168.6.5+62857 AS 17
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.163.6.4     Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
```

```
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            6
    Accepted prefixes:            1
    Suppressed due to damping:    0
    Advertised prefixes:          6
  Last traffic (seconds): Received 5    Sent 3    Checked 19
  Input messages:  Total 2961   Updates 7       Refreshes 0     Octets 56480
  Output messages: Total 2945   Updates 6       Refreshes 0     Octets 56235
  Output Queue[0]: 0

Peer: 192.168.0.1+179 AS 17    Local: 192.168.6.5+60068 AS 17
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.0.1     Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 3
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            6
    Accepted prefixes:            1
    Suppressed due to damping:    0
    Advertised prefixes:          6
  Last traffic (seconds): Received 18   Sent 20   Checked 12
  Input messages:  Total 15     Updates 5       Refreshes 0     Octets 447
  Output messages: Total 554    Updates 4       Refreshes 0     Octets 32307
```

```
  Output Queue[0]: 0

Peer: 192.168.5.5+57458 AS 17  Local: 192.168.6.5+179 AS 17
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.5.5     Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 2
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            7
    Accepted prefixes:            7
    Suppressed due to damping:    0
    Advertised prefixes:          6
  Last traffic (seconds): Received 17   Sent 3    Checked 9
  Input messages:  Total 2967   Updates 7       Refreshes 0     Octets 56629
  Output messages: Total 2943   Updates 6       Refreshes 0     Octets 56197
  Output Queue[0]: 0

Peer: 192.168.40.4+53990 AS 17 Local: 192.168.6.5+179 AS 17
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ send-ospf ]
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 192.168.6.5 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.40.4    Local ID: 192.168.6.5       Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 1
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
```

```
  Peer supports 4 byte AS extension (peer-as 17)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            7
    Accepted prefixes:            7
    Suppressed due to damping:    0
    Advertised prefixes:          6
  Last traffic (seconds): Received 5    Sent 23   Checked 52
  Input messages:  Total 2960   Updates 7       Refreshes 0     Octets 56496
  Output messages: Total 2943   Updates 6       Refreshes 0     Octets 56197
  Output Queue[0]: 0
```

Verifying BGP Groups

Purpose Verify that the BGP groups are configured correctly.

Action From operational mode, enter the **show bgp group** command.

```
user@A> show bgp group
Group Type: Internal    AS: 17                    Local AS: 17
  Name: internal-peers  Index: 0                  Flags: <>
  Export: [ send-ospf ]
  Options: <Cluster>
  Holdtime: 0
  Total peers: 4        Established: 4
  192.163.6.4+179
  192.168.40.4+53990
  192.168.0.1+179
  192.168.5.5+57458
  inet.0: 0/26/16/0

Groups: 1  Peers: 4    External: 0    Internal: 4    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                26          0          0          0          0          0
```

Verifying BGP Summary Information

Purpose Verify that the BGP configuration is correct.

Action From operational mode, enter the **show bgp summary** command.

```
user@A> show bgp summary

Groups: 1 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                26          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.163.6.4              17       2981       2965       0       0    22:19:15 0/6/1/0              0/0/0/0
192.168.0.1              17         36        575       0       0       13:43 0/6/1/0              0/0/0/0
192.168.5.5              17       2988       2964       0       0    22:19:10 0/7/7/0              0/0/0/0
192.168.40.4             17       2980       2964       0       0    22:19:14 0/7/7/0              0/0/0/0
```

Verifying Routing Table Information

Purpose Verify that the routing table contains the IBGP routes.

Action From operational mode, enter the **show route** command.

```
user@A> show route
inet.0: 12 destinations, 38 routes (12 active, 0 holddown, 10 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.0/30      *[Direct/0] 22:22:03
                    > via fe-0/0/0.1
                    [BGP/170] 22:20:55, MED 2, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:51, MED 3, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
10.10.10.1/32      *[Local/0] 22:22:03
                      Local via fe-0/0/0.1
10.10.10.4/30      *[OSPF/10] 22:21:13, metric 2
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:51, MED 4, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
10.10.10.8/30      *[Direct/0] 22:22:03
                    > via fe-0/0/1.3
                    [BGP/170] 22:20:51, MED 2, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:55, MED 3, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
10.10.10.9/32      *[Local/0] 22:22:03
                      Local via fe-0/0/1.3
10.10.10.12/30     *[OSPF/10] 22:21:08, metric 2
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:55, MED 4, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
192.163.6.4/32     *[OSPF/10] 22:21:13, metric 1
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:55, MED 1, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:51, MED 3, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
192.168.0.1/32     *[OSPF/10] 22:21:08, metric 1
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:51, MED 1, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:55, MED 3, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
192.168.5.5/32     *[OSPF/10] 22:21:08, metric 2
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 00:15:24, MED 1, localpref 100, from 192.168.0.1
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:55, MED 4, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
192.168.6.5/32     *[Direct/0] 22:22:04
```

```
                    > via lo0.1
                    [BGP/170] 22:20:51, MED 2, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
                    [BGP/170] 22:20:55, MED 2, localpref 100, from 192.168.40.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
192.168.40.4/32    *[OSPF/10] 22:21:13, metric 2
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:55, MED 1, localpref 100, from 192.163.6.4
                      AS path: I
                    > to 10.10.10.2 via fe-0/0/0.1
                    [BGP/170] 22:20:51, MED 4, localpref 100, from 192.168.5.5
                      AS path: I
                    > to 10.10.10.10 via fe-0/0/1.3
224.0.0.5/32       *[OSPF/10] 22:22:07, metric 1
                      MultiRecv
```

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

Example: Configuring BGP Confederations

- Understanding BGP Confederations on page 292
- Example: Configuring BGP Confederations on page 293

Understanding BGP Confederations

BGP confederations are another way to solve the scaling problems created by the BGP full mesh requirement. BGP confederations effectively break up a large autonomous system (AS) into subautonomous systems (sub-ASs). Each sub-AS must be uniquely identified within the confederation AS by a sub-AS number. Typically, sub-AS numbers are taken from the private AS numbers between 64,512 and 65,535.

Within a sub-AS, the same internal BGP (IBGP) full mesh requirement exists. Connections to other confederations are made with standard external BGP (EBGP), and peers outside the sub-AS are treated as external. To avoid routing loops, a sub-AS uses a confederation sequence, which operates like an AS path but uses only the privately assigned sub-AS numbers.

The confederation AS appears whole to other confederation ASs. The AS path received by other ASs shows only the globally assigned AS number. It does not include the confederation sequence or the privately assigned sub-AS numbers. The sub-AS numbers are removed when the route is advertised out of the confederation AS.
Figure 30 on page 293 shows an AS divided into four confederations.

Figure 30: BGP Confederations

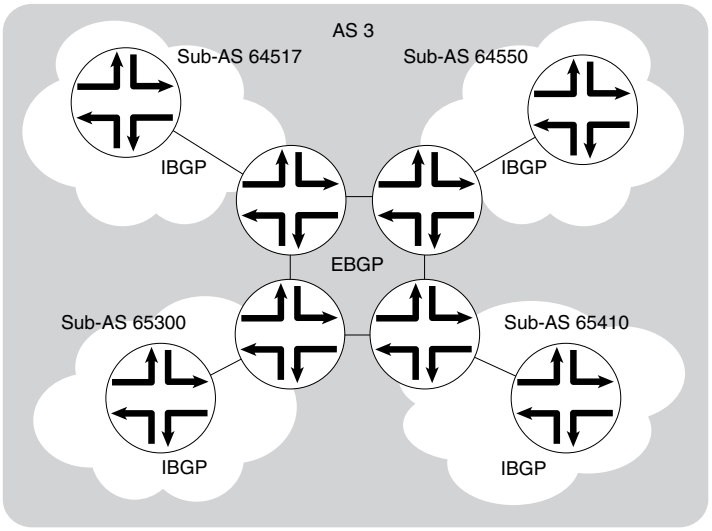


Figure 30 on page 293 shows AS 3 divided into four sub-ASs, 64517, 64550, 65300, and 65410, which are linked through EBGP sessions. Because the confederations are connected by EBGP, they do not need to be fully meshed. EBGP routes are readvertised to other sub-ASs.

Example: Configuring BGP Confederations

This example shows how to configure BGP confederations.

Requirements

- Configure network interfaces.
- Configure external peer sessions. See "Example: Configuring External BGP Point-to-Point Peer Sessions" on page 12.
- Configure interior gateway protocol (IGP) sessions between peers.
- Configure a routing policy to advertise the BGP routes.

Overview

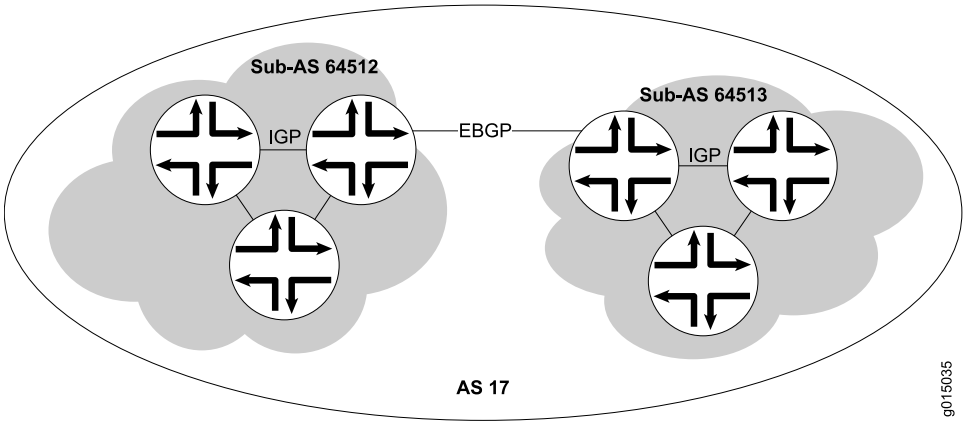
Within a BGP confederation, the links between the confederation member autonomous systems (ASs) must be external BGP (EBGP) links, not internal BGP (IBGP) links.

Similar to route reflectors, BGP confederations reduce the number of peer sessions and TCP sessions to maintain connections between IBGP routing devices. BGP confederation is one method used to solve the scaling problems created by the IBGP full mesh

requirement. BGP confederations effectively break up a large AS into subautonomous systems. Each sub-AS must be uniquely identified within the confederation AS by a sub-AS number. Typically, sub-AS numbers are taken from the private AS numbers between 64512 and 65535. Within a sub-AS, the same IBGP full mesh requirement exists. Connections to other confederations are made with standard EBGP, and peers outside the sub-AS are treated as external. To avoid routing loops, a sub-AS uses a confederation sequence, which operates like an AS path but uses only the privately assigned sub-AS numbers.

Figure 31 on page 294 shows a sample network in which AS 17 has two separate confederations: sub-AS 64512 and sub-AS 64513, each of which has multiple routers. Within a sub-AS, an IGP is used to establish network connectivity with internal peers. Between sub-ASs, an EBGP peer session is established.

Figure 31: Typical Network Using BGP Confederations



Configuration

CLI Quick Configuration

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

All Devices in Sub-AS 64512

```
set routing-options autonomous-system 64512
set routing-options confederation 17 members 64512
set routing-options confederation 17 members 64513
set protocols bgp group sub-AS-64512 type internal
set protocols bgp group sub-AS-64512 local-address 192.168.5.1
set protocols bgp group sub-AS-64512 neighbor 192.168.8.1
set protocols bgp group sub-AS-64512 neighbor 192.168.15.1
```

Border Device in Sub-AS 64512

```
set protocols bgp group to-sub-AS-64513 type external
set protocols bgp group to-sub-AS-64513 peer-as 64513
set protocols bgp group to-sub-AS-64513 neighbor 192.168.5.2
```

All Devices in Sub-AS 64513

```
set routing-options autonomous-system 64513
set routing-options confederation 17 members 64512
set routing-options confederation 17 members 64513
set protocols bgp group sub-AS-64513 type internal
```

```
set protocols bgp group sub-AS-64513 local-address 192.168.5.2
set protocols bgp group sub-AS-64513 neighbor 192.168.9.1
set protocols bgp group sub-AS-64513 neighbor 192.168.16.1
```

Border Device in Sub-AS 64513

```
set protocols bgp group to-sub-AS-64512 type external
set protocols bgp group to-sub-AS-64512 peer-as 64512
set protocols bgp group to-sub-AS-64512 neighbor 192.168.5.1
```

Step-by-Step Procedure

This procedure shows the steps for the devices that are in sub-AS 64512.

The **autonomous-system** statement sets the sub-AS number of the device.

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

To configure BGP confederations:

1. Set the sub-AS number for the device.

   ```
   [edit routing-options]
   user@host# set autonomous-system 64512
   ```

2. In the confederation, include all sub-ASs in the main AS.

   The number 17 represents the main AS. The **members** statement lists all the sub-ASs in the main AS.

   ```
   [edit routing-options confederation]
   user@host# set 17 members 64512
   user@host# set 17 members 64513
   ```

3. On the border device in sub-AS 64512, configure an EBGP connection to the border device in AS 64513.

   ```
   [edit protocols bgp group to-sub-AS-64513]
   user@host# set type external
   user@host# set neighbor 192.168.5.2
   user@host# set peer-as 64513
   ```

4. Configure an IBGP group for peering with the devices within sub-AS 64512.

   ```
   [edit protocols bgp group sub-AS-64512]
   user@host# set type internal
   user@host# set local-address 192.168.5.1
   user@host# neighbor 192.168.8.1
   user@host# neighbor 192.168.15.1
   ```

Results

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

```
user@host# show routing-options
autonomous-system 64512;
confederation 17 members [ 64512 64513 ];

user@host# show protocols
bgp {
```

```
    group to-sub-AS-64513 { # On the border devices only
      type external;
      peer-as 64513;
      neighbor 192.168.5.2;
    }
    group sub-AS-64512 {
      type internal;
      local-address 192.168.5.1;
      neighbor 192.168.8.1;
      neighbor 192.168.15.1;
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode. Repeat these steps for sSub-AS 64513.

Verification

Confirm that the configuration is working properly.

- Verifying BGP Neighbors on page 296
- Verifying BGP Groups on page 297
- Verifying BGP Summary Information on page 298

Verifying BGP Neighbors

Purpose Verify that BGP is running on configured interfaces and that the BGP session is active for each neighbor address.

Action From the CLI, enter the **show bgp neighbor** command.

Sample Output

```
user@host> show bgp neighbor
Peer: 10.255.245.12+179 AS 35  Local: 10.255.245.13+2884 AS 35
  Type: Internal    State: Established  (route reflector client)Flags: Sync
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: Preference LocalAddress HoldTime Cluster AddressFamily Rib-group Refresh

  Address families configured: inet-vpn-unicast inet-labeled-unicast
  Local Address: 10.255.245.13 Holdtime: 90 Preference: 170
  Flags for NLRI inet-vpn-unicast: AggregateLabel
  Flags for NLRI inet-labeled-unicast: AggregateLabel
  Number of flaps: 0
  Peer ID: 10.255.245.12    Local ID: 10.255.245.13    Active Holdtime: 90
  Keepalive Interval: 30
  NLRI advertised by peer: inet-vpn-unicast inet-labeled-unicast
  NLRI for this session: inet-vpn-unicast inet-labeled-unicast
  Peer supports Refresh capability (2)
Restart time configured on the peer: 300
  Stale routes from peer are kept for: 60
  Restart time requested by this peer: 300
  NLRI that peer supports restart for: inet-unicast inet6-unicast
  NLRI that restart is negotiated for: inet-unicast inet6-unicast
  NLRI of received end-of-rib markers: inet-unicast inet6-unicast
  NLRI of all end-of-rib markers sent: inet-unicast inet6-unicast
```

```
  Table inet.0 Bit: 10000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 4
    Received prefixes: 6
    Suppressed due to damping: 0
  Table inet6.0 Bit: 20000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 0
    Received prefixes: 2
    Suppressed due to damping: 0
  Last traffic (seconds): Received 3    Sent 3    Checked 3
  Input messages:  Total 9      Updates 6       Refreshes 0     Octets 403
  Output messages: Total 7      Updates 3       Refreshes 0     Octets 365
  Output Queue[0]: 0
  Output Queue[1]: 0
  Trace options: detail packets
  Trace file: /var/log/bgpgr size 131072 files 10
```

Meaning The output shows a list of the BGP neighbors with detailed session information. Verify the following information:

- Each configured peering neighbor is listed.
- For **State**, each BGP session is **Established**.
- For **Type**, each peer is configured as the correct type (either internal or external).
- For **AS**, the AS number of the BGP neighbor is correct.

Verifying BGP Groups

Purpose Verify that the BGP groups are configured correctly.

Action From the CLI, enter the **show bgp group** command.

Sample Output

```
user@host> show bgp group
Group Type: Internal    AS: 10045       Local AS: 10045
  Name: pe-to-asbr2                                 Flags: Export Eval
  Export: [ match-all ]
  Total peers: 1        Established: 1
  10.0.0.4+179
  bgp.l3vpn.0: 1/1/0
  vpn-green.inet.0: 1/1/0

Groups: 1   Peers: 1    External: 0    Internal: 1    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            1          1          0          0          0          0
```

Meaning The output shows a list of the BGP groups with detailed group information. Verify the following information:

- Each configured group is listed.
- For **AS**, each group's remote AS is configured correctly.

- For **Local AS**, each group's local AS is configured correctly.
- For **Group Type**, each group has the correct type (either internal or external).
- For **Total peers**, the expected number of peers within the group is shown.
- For **Established**, the expected number of peers within the group have BGP sessions in the **Established** state.
- The IP addresses of all the peers within the group are present.

Verifying BGP Summary Information

Purpose Verify that the BGP configuration is correct.

Action From the CLI, enter the **show bgp summary** command.

Sample Output

```
user@host> show bgp summary
Groups: 1 Peers: 3 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 6          4          0          0          0          0
Peer                 AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.0.0.2           65002      88675      88652       0       2       42:38 2/4/0
                  0/0/0
10.0.0.3           65002      54528      54532       0       1     2w4d22h 0/0/0
                  0/0/0
10.0.0.4           65002      51597      51584       0       0     2w3d22h 2/2/0
                  0/0/0
```

Meaning The output shows a summary of BGP session information. Verify the following information:

- For **Groups**, the total number of configured groups is shown.
- For **Peers**, the total number of BGP peers is shown.
- For **Down Peers**, the total number of unestablished peers is 0. If this value is not zero, one or more peering sessions are not yet established.
- Under **Peer**, the IP address for each configured peer is shown.
- Under **AS**, the peer AS for each configured peer is correct.
- Under **Up/Dwn State**, the BGP state reflects the number of paths received from the neighbor, the number of these paths that have been accepted, and the number of routes being damped (such as 0/0/0). If the field is **Active**, it indicates a problem in the establishment of the BGP session.

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

CHAPTER 8

BGP Security Configuration

Example: Configuring BGP Route Authentication

Understanding Route Authentication

The use of router and route authentication and route integrity greatly mitigates the risk of being attacked by a machine or router that has been configured to share incorrect routing information with another router. In this kind of attack, the attacked router can be tricked into creating a routing loop, or the attacked router's routing table can be greatly increased thus impacting performance, or routing information can be redirected to a place in the network for the attacker to analyze it. Bogus route advertisements can be sent out on a segment. These updates can be accepted into the routing tables of neighbor routers unless an authentication mechanism is in place to verify the source of the routes.

Router and route authentication enables routers to share information only if they can verify that they are talking to a trusted source, based on a password (key). In this method, a hashed key is sent along with the route being sent to another router. The receiving router compares the sent key to its own configured key. If they are the same, it accepts the route. By using a hashing algorithm, the key is not sent over the wire in plain text. Instead, a hash is calculated using the configured key. The routing update is used as the input text, along with the key, into the hashing function. This hash is sent along with the route update to the receiving router. The receiving router compares the received hash with a hash it generates on the route update using the preshared key configured on it. If the two hashes are the same, the route is assumed to be from a trusted source. The key is known only to the sending and receiving routers.

To further strengthen security, you can configure a series of authentication keys (a *keychain*). Each key has a unique start time within the keychain. Keychain authentication allows you to change the password information periodically without bringing down peering sessions. This keychain authentication method is referred to as *hitless* because the keys roll over from one to the next without resetting any peering sessions or interrupting the routing protocol.

The sending peer uses the following rules to identify the active authentication key:

- The start time is less than or equal to the current time (in other words, not in the future).
- The start time is greater than that of all other keys in the chain whose start time is less than the current time (in other words, closest to the current time).

The receiving peer determines the key with which it authenticates based on the incoming key identifier.

The sending peer identifies the current authentication key based on a configured start time and then generates a hash value using the current key. The sending peer then inserts a TCP-enhanced authentication option object into the BGP update message. The object contains an object ID (assigned by IANA), the object length, the current key, and a hash value.

The receiving peer examines the incoming TCP-enhanced authentication option, looks up the received authentication key, and determines whether the key is acceptable based on the start time, the system time, and the tolerance parameter. If the key is accepted, the receiving peer calculates a hash and authenticates the update message.

Initial application of a keychain to a TCP session causes the session to reset. However, once the keychain is applied, the addition or removal of a password from the keychain does not cause the TCP session to reset. Also, the TCP session does not reset when the keychain changes from one authentication algorithm to another.

Example: Configuring Route Authentication for BGP

All BGP protocol exchanges can be authenticated to guarantee that only trusted routing devices participate in autonomous system (AS) routing updates. By default, authentication is disabled.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol (IGP).

Overview

When you configure authentication, the algorithm creates an encoded checksum that is included in the transmitted packet. The receiving routing device uses an authentication key (password) to verify the packet's checksum.

This example includes the following statements for configuring and applying the keychain:

- **key**—A keychain can have multiple keys. Each key within a keychain must be identified by a unique integer value. The range of valid identifier values is from 0 through 63.

  The key can be up to 126 characters long. Characters can include any ASCII strings. If you include spaces, enclose all characters in quotation marks (“ ”).

- **tolerance**—(Optional) For each keychain, you can configure a clock-skew tolerance value in seconds. The clock-skew tolerance is applicable to the receiver accepting keys for BGP updates. The configurable range is 0 through 999,999,999 seconds. During the tolerance period, either the current or previous password is acceptable.

- **key-chain**—For each keychain, you must specify a name. This example defines one keychain: **bgp-auth**. You can have multiple keychains on a routing device. For example, you can have a keychain for BGP, a keychain for OSPF, and a keychain for LDP.

- **secret**—For each key in the keychain, you must set a secret password. This password can be entered in either encrypted or plain text format in the **secret** statement. It is always displayed in encrypted format.

- **start-time**—Each key must specify a start time in UTC format. Control gets passed from one key to the next. When a configured start time arrives (based on the routing device’s clock), the key with that start time becomes active. Start times are specified in the local time zone for a routing device and must be unique within the keychain.

- **authentication-key-chain**—Enables you to apply a keychain at the global BGP level for all peers, for a group, or for a neighbor. This example applies the keychain to the peers defined in the external BGP (EBGP) group called **ext**.

- **authentication-algorithm**—For each keychain, you can specify a hashing algorithm. The algorithm can be AES-128, MD5, or SHA-1.

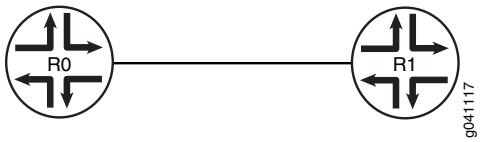
  You associate a keychain and an authentication algorithm with a BGP neighboring session.

This example configures a keychain named **bgp-auth**. Key 0 will be sent and accepted starting at 2011-6-23.20:19:33 -0700, and will stop being sent and accepted when the next key in the keychain (key 1) becomes active. Key 1 becomes active one year later at 2012-6-23.20:19:33 -0700, and will not stop being sent and accepted unless another key is configured with a start time that is later than the start time of key 1. A clock-skew tolerance of 30 seconds applies to the receiver accepting the keys. During the tolerance period, either the current or previous key is acceptable. The keys are shared-secret passwords. This means that the neighbors receiving the authenticated routing updates must have the same authentication keychain configuration, including the same keys (passwords). So Router R0 and Router R1 must have the same authentication-key-chain configuration if they are configured as peers. This example shows the configuration on only one of the routing devices.

Topology Diagram

Figure 32 on page 302 shows the topology used in this example.

Figure 32: Authentication for BGP



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 protocols bgp group ext type external
set protocols bgp group ext peer-as 65530
set protocols bgp group ext neighbor 172.16.2.1
set routing-options autonomous-system 65533
set protocols bgp group ext authentication-key-chain bgp-auth
set protocols bgp group ext authentication-algorithm md5
set security authentication-key-chains key-chain bgp-auth tolerance 30
set security authentication-key-chains key-chain bgp-auth key 0 secret
  this-is-the-secret-password
set security authentication-key-chains key-chain bgp-auth key 0 start-time
  2011-6-23.20:19:33-0700
set security authentication-key-chains key-chain bgp-auth key 1 secret
  this-is-another-secret-password
set security authentication-key-chains key-chain bgp-auth key 1 start-time
  2012-6-23.20:19:33-0700
```

Step-by-Step Procedure

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

To configure Router R1 to accept route filters from Device CE1 and perform outbound route filtering using the received filters:

1. Configure the local autonomous system.

   ```
   [edit routing-options]
   user@R1# set autonomous-system 65533
   ```

2. Configure one or more BGP groups.

   ```
   [edit protocols bgp group ext]
   user@R1# set type external
   user@R1# set peer-as 65530
   user@R1# set neighbor 172.16.2.1
   ```

3. Configure authentication with multiple keys.

   ```
   [edit security authentication-key-chains key-chain bgp-auth]
   user@R1# set key 0 secret this-is-the-secret-password
   user@R1# set key 0 start-time 2011-6-23.20:19:33-0700
   user@R1# set key 1 secret this-is-another-secret-password
   user@R1# set key 1 start-time 2012-6-23.20:19:33-0700
   ```

The start time of each key must be unique within the keychain.

4. Apply the authentication keychain to BGP, and set the hashing algorithm.

```
[edit protocols bgp group ext]
user@R1# set authentication-key-chain bgp-auth
user@R1# set authentication-algorithm md5
```

5. (Optional) Apply a clock-skew tolerance value in seconds.

```
[edit security authentication-key-chains key-chain bgp-auth]
user@R1# set tolerance 30
```

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

```
user@R1# show protocols
bgp {
  group ext {
    type external;
    peer-as 65530;
    neighbor 172.16.2.1;
    authentication-key-chain bgp-auth;
    authentication-algorithm md5;
  }
}

user@R1# show routing-options
autonomous-system 65533;

user@R1# show security
authentication-key-chains {
  key-chain bgp-auth {
    tolerance 30;
      key 0 {
      secret
        "$9$5T6AREylK8RhxNdwaJn/CtO1cyKvWx9AylMWdVgoJDjqP5FCA0z3lEhcMWLxNbgJDiF6A";
        ## SECRET-DATA
      start-time “2011-6-23.20:19:33 -0700”;
    }
    key 1 {
      secret "$9$UyD.59CuO1h9AylKW-dqmfT369CuRhSP5hrvMN-JGDiqfu0IleWpuh.";
        ## SECRET-DATA
      start-time “2012-6-23.20:19:33 -0700”;
    }
  }
}
```

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

Repeat the procedure for every BGP-enabled device in the network, using the appropriate interface names and addresses for each BGP-enabled device.

Verification

Confirm that the configuration is working properly.

Verifying Authentication for the Neighbor

Purpose Make sure that the **AutheKeyChain** option appears in the output of the **show bgp neighbor** command.

Action From operational mode, enter the show bgp neighbor command.

```
user@R1> show bgp neighbor
Peer: 172.16.2.1+179 AS 65530  Local: 172.16.2.2+1222 AS 65533
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ direct-lo0 ]
  Options: <Preference PeerAS Refresh>
  Options: <AutheKeyChain>
  Authentication key is configured
  Authentication key chain: jni
  Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 172.16.2.1       Local ID: 10.255.124.35    Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  Local Interface: fe-0/0/1.0
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              2
    Received prefixes:            2
    Suppressed due to damping:    0
    Advertised prefixes:          1
  Last traffic (seconds): Received 2    Sent 2    Checked 2
  Input messages:  Total 21     Updates 2       Refreshes 0     Octets 477
  Output messages: Total 22     Updates 1       Refreshes 0     Octets 471
  Output Queue[0]: 0
```

Verifying That Authorization Messages Are Sent

Purpose Confirm that BGP has the enhanced authorization option.

Action From operational mode, enter the **monitor traffic interface fe-0/0/1** command.

```
user@R1> monitor traffic interface fe-0/0/1
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Listening on fe-0/0/1, capture size 96 bytes

13:08:00.618402  In arp who-has 172.16.2.66 tell 172.16.2.69
```

```
13:08:02.408249 Out IP 172.16.2.2.1122 > 172.16.2.1.646: P
1889289217:1889289235(18) ack 2215740969 win 58486 <nop,nop,timestamp 167557
1465469,nop,Enhanced Auth keyid 0 diglen 12 digest: fe3366001f45767165f17037>:
13:08:02.418396  In IP 172.16.2.1.646 > 172.16.2.2.1122: P 1:19(18) ack 18 win
57100 <nop,nop,timestamp 1466460 167557,nop,Enhanced Auth keyid 0 diglen 12
digest: a18c31eda1b14b2900921675>:
13:08:02.518146 Out IP 172.16.2.2.1122 > 172.16.2.1.646: . ack 19 win 58468
<nop,nop,timestamp 167568 1466460,nop,Enhanced Auth keyid 0 diglen 12 digest:
c3b6422eb6bd3fd9cf79742b>
13:08:28.199557 Out IP 172.16.2.2.nerv > 172.16.2.1.bgp: P
286842489:286842508(19) ack 931203976 win 57200 <nop,Enhanced Auth keyid 0
diglen 12 digest: fc0e42900a73736bcc07c1a4>: BGP, length: 19
13:08:28.209661  In IP 172.16.2.1.bgp > 172.16.2.2.nerv: P 1:20(19) ack 19 win
56835 <nop,Enhanced Auth keyid 0 diglen 12 digest: 0fc8578c489fabce63aeb2c3>:
BGP, length: 19
13:08:28.309525 Out IP 172.16.2.2.nerv > 172.16.2.1.bgp: . ack 20 win 57181
<nop,Enhanced Auth keyid 0 diglen 12 digest: ef03f282fb2ece0039491df8>
13:08:32.439708 Out IP 172.16.2.2.1122 > 172.16.2.1.646: P 54:72(18) ack 55 win
58432 <nop,nop,timestamp 170560 1468472,nop,Enhanced Auth keyid 0 diglen 12
digest: 76e0cf926f348b726c631944>:
13:08:32.449795  In IP 172.16.2.1.646 > 172.16.2.2.1122: P 55:73(18) ack 72 win
57046 <nop,nop,timestamp 1469463 170560,nop,Enhanced Auth keyid 0 diglen 12
digest: dae3eec390d18a114431f4d8>:
13:08:32.549726 Out IP 172.16.2.2.1122 > 172.16.2.1.646: . ack 73 win 58414
<nop,nop,timestamp 170571 1469463,nop,Enhanced Auth keyid 0 diglen 12 digest:
851df771aee2ea7a43a0c46c>
13:08:33.719880  In arp who-has 172.16.2.66 tell 172.16.2.69
^C
35 packets received by filter
0 packets dropped by kernel
```

Checking Authentication Errors

Purpose Check the number of packets dropped by TCP because of authentication errors.

Action From operational mode, enter the **show system statistics tcp | match auth** command.

```
user@R1> show system statistics tcp | match auth
        0 send packets dropped by TCP due to auth errors
        58 rcv packets dropped by TCP due to auth errors
```

Verifying the Operation of the Keychain

Purpose Check the number of packets dropped by TCP because of authentication errors.

Action From operational mode, enter the **show security keychain detail** command.

```
user@R1> show security keychain detail
keychain                Active-ID       Next-ID       Transition  Tolerance
                      Send  Receive   Send  Receive
 bgp-auth               3     3        1     1        1d 23:58    30
  Id 3, Algorithm hmac-md5, State send-receive, Option basic
  Start-time Wed Aug 11 16:28:00 2010, Mode send-receive
  Id 1, Algorithm hmac-md5, State inactive, Option basic
  Start-time Fri Aug 20 11:30:57 2010, Mode send-receive
```

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

Examples: Configuring TCP and BGP Security

Understanding Security Options for BGP with TCP

Among routing protocols, BGP is unique in using TCP as its transport protocol. BGP peers are established by manual configuration between routing devices to create a TCP session on port 179. A BGP-enabled device periodically sends keepalive messages to maintain the connection.

Over time, BGP has become the dominant interdomain routing protocol on the Internet. However, it has limited guarantees of stability and security. Configuring security options for BGP must balance suitable security measures with acceptable costs. No one method has emerged as superior to other methods. Each network administrator must configure security measures that meet the needs of the network being used.

For detailed information about the security issues associated with BGP's use of TCP as a transport protocol, see RFC 4272, *BGP Security Vulnerabilities Analysis*.

Example: Configuring a Filter to Block TCP Access to a Port Except from Specified BGP Peers

This example shows how to configure a standard stateless firewall filter that blocks all TCP connection attempts to port 179 from all requesters except from specified BGP peers.

Requirements

No special configuration beyond device initialization is required before you configure this example.

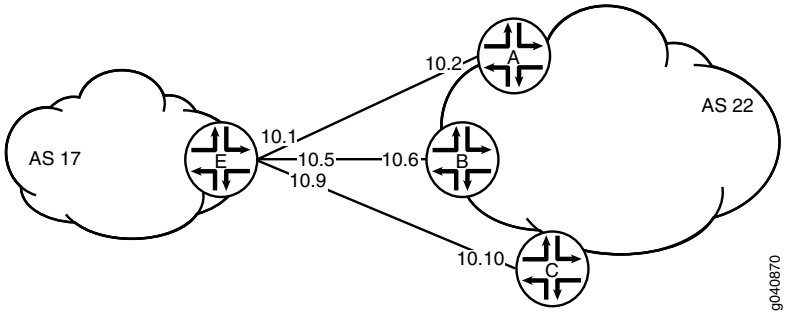
Overview

In this example, you create a stateless firewall filter that blocks all TCP connection attempts to port 179 from all requesters except the specified BGP peers.

The stateless firewall filter **filter_bgp179** matches all packets from the directly connected interfaces on Device A and Device B to the destination port number 179.

Figure 33 on page 307 shows the topology used in this example. Device C attempts to make a TCP connection to Device E. Device E blocks the connection attempt. This example shows the configuration on Device E.

Figure 33: Typical Network with BGP Peer Sessions



Configuration

CLI Quick Configuration

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

Device C

```
set interfaces ge-1/2/0 unit 10 description to-E
set interfaces ge-1/2/0 unit 10 family inet address 10.10.10.10/30
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 17
set protocols bgp group external-peers neighbor 10.10.10.9
set routing-options autonomous-system 22
```

Device E

```
set interfaces ge-1/2/0 unit 0 description to-A
set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/30
set interfaces ge-1/2/1 unit 5 description to-B
set interfaces ge-1/2/1 unit 5 family inet address 10.10.10.5/30
set interfaces ge-1/0/0 unit 9 description to-C
set interfaces ge-1/0/0 unit 9 family inet address 10.10.10.9/30
set interfaces lo0 unit 2 family inet filter input filter_bgp179
set interfaces lo0 unit 2 family inet address 192.168.0.1/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 22
set protocols bgp group external-peers neighbor 10.10.10.2
set protocols bgp group external-peers neighbor 10.10.10.6
set protocols bgp group external-peers neighbor 10.10.10.10
set routing-options autonomous-system 17
set firewall family inet filter filter_bgp179 term 1 from source-address 10.10.10.2/32
set firewall family inet filter filter_bgp179 term 1 from source-address 10.10.10.6/32
set firewall family inet filter filter_bgp179 term 1 from destination-port bgp
set firewall family inet filter filter_bgp179 term 1 then accept
set firewall family inet filter filter_bgp179 term 2 then reject
```

Configuring Device E

Step-by-Step Procedure

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

To configure Device E with a stateless firewall filter that blocks all TCP connection attempts to port 179 from all requestors except specified BGP peers:

1. Configure the interfaces.

```
user@E# set interfaces ge-1/2/0 unit 0 description to-A
user@E# set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/30

user@E# set interfaces ge-1/2/1 unit 5 description to-B
user@E# set interfaces ge-1/2/1 unit 5 family inet address 10.10.10.5/30

user@E# set interfaces ge-1/0/0 unit 9 description to-C
user@E# set interfaces ge-1/0/0 unit 9 family inet address 10.10.10.9/30
```

2. Configure BGP.

```
[edit protocols bgp group external-peers]
user@E# set type external
user@E# set peer-as 22
user@E# set neighbor 10.10.10.2
user@E# set neighbor 10.10.10.6
user@E# set neighbor 10.10.10.10
```

3. Configure the autonomous system number.

```
[edit routing-options]
user@E# set autonomous-system 17
```

4. Define the filter term that accepts TCP connection attempts to port 179 from the specified BGP peers.

```
[edit firewall family inet filter filter_bgp179]
user@E# set term 1 from source-address 10.10.10.2/32
user@E# set term 1 from source-address 10.10.10.6/32
user@E# set term 1 from destination-port bgp
user@E# set term 1 then accept
```

5. Define the other filter term to reject packets from other sources.

```
[edit firewall family inet filter filter_bgp179]
user@E# set term 2 then reject
```

6. Apply the firewall filter to the loopback interface.

```
[edit interfaces lo0 unit 2 family inet]
user@E# set filter input filter_bgp179
user@E# set address 192.168.0.1/32
```

Results

From configuration mode, confirm your configuration by entering the **show firewall**, **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not

display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@E# show firewall
family inet {
    filter filter_bgp179 {
        term 1 {
            from {
                source-address {
                    10.10.10.2/32;
                    10.10.10.6/32;
                }
                destination-port bgp;
            }
            then accept;
        }
        term 2 {
            then {
                reject;
            }
        }
    }
}
user@E# show interfaces
lo0 {
    unit 2 {
        family inet {
            filter {
                input filter_bgp179;
            }
            address 192.168.0.1/32;
        }
    }
}
ge-1/2/0 {
    unit 0 {
        description to-A;
        family inet {
            address 10.10.10.1/30;
        }
    }
}
ge-1/2/1 {
    unit 5 {
        description to-B;
        family inet {
            address 10.10.10.5/30;
        }
    }
}
ge-1/0/0 {
    unit 9 {
        description to-C;
        family inet {
            address 10.10.10.9/30;
```

```
    }
  }
}
```

```
user@E# show protocols
bgp {
  group external-peers {
    type external;
    peer-as 22;
    neighbor 10.10.10.2;
    neighbor 10.10.10.6;
    neighbor 10.10.10.10;
  }
}
```

```
user@E# show routing-options
autonomous-system 17;
```

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

Verification

Confirm that the configuration is working properly.

- Verifying That the Filter Is Configured on page 310
- Verifying the TCP Connections on page 310
- Monitoring Traffic on the Interfaces on page 311

Verifying That the Filter Is Configured

Purpose Make sure that the filter is listed in output of the **show firewall filter** command.

Action

```
user@E> show firewall filter filter_bgp179
Filter: filter_bgp179
```

Verifying the TCP Connections

Purpose Verify the TCP connections.

Action From operational mode, run the **show system connections extensive** command on Device C and Device E.

The output on Device C shows the attempt to establish a TCP connection. The output on Device E shows that connections are established with Device A and Device B only.

```
user@C> show system connections extensive | match 10.10.10

tcp4       0      0  10.10.10.9.51872     10.10.10.10.179        SYN_SENT
```

```
user@E> show system connections extensive | match 10.10.10

tcp4       0      0  10.10.10.5.179       10.10.10.6.62096       ESTABLISHED
tcp4       0      0  10.10.10.6.62096     10.10.10.5.179         ESTABLISHED
tcp4       0      0  10.10.10.1.179       10.10.10.2.61506       ESTABLISHED
tcp4       0      0  10.10.10.2.61506     10.10.10.1.179         ESTABLISHED
```

Monitoring Traffic on the Interfaces

Purpose Use the **monitor traffic** command to compare the traffic on an interface that establishes a TCP connection with the traffic on an interface that does not establish a TCP connection.

Action From operational mode, run the **monitor traffic** command on the Device E interface to Device B and on the Device E interface to Device C. The following sample output verifies that in the first example, acknowledgment (**ack**) messages are received. In the second example, **ack** messages are not received.

```
user@E> monitor traffic size 1500 interface ge-1/2/1.5
19:02:49.700912 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: P
3330573561:3330573580(19) ack 915601686 win 16384 <nop,nop,timestamp 1869518816
1869504850>: BGP, length: 19
19:02:49.801244  In IP 10.10.10.6.62096 > 10.10.10.5.bgp: . ack 19 win 16384
<nop,nop,timestamp 1869518916 1869518816>
19:03:03.323018  In IP 10.10.10.6.62096 > 10.10.10.5.bgp: P 1:20(19) ack 19 win
16384 <nop,nop,timestamp 1869532439 1869518816>: BGP, length: 19
19:03:03.422418 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: . ack 20 win 16384
<nop,nop,timestamp 1869532539 1869532439>
19:03:17.220162 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: P 19:38(19) ack 20 win
 16384 <nop,nop,timestamp 1869546338 1869532439>: BGP, length: 19
19:03:17.320501  In IP 10.10.10.6.62096 > 10.10.10.5.bgp: . ack 38 win 16384
<nop,nop,timestamp 1869546438 1869546338>

user@E> monitor traffic size 1500 interface ge-1/0/0.9

18:54:20.175471 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
 win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869009240 0,sackOK,eol>
18:54:23.174422 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
 win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869012240 0,sackOK,eol>
18:54:26.374118 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
 win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869015440 0,sackOK,eol>
18:54:29.573799 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
 win 16384 <mss 1460,sackOK,eol>
18:54:32.773493 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
 win 16384 <mss 1460,sackOK,eol>
18:54:35.973185 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
 win 16384 <mss 1460,sackOK,eol>
```

Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List

This example shows how to configure a standard stateless firewall filter that limits certain TCP and Internet Control Message Protocol (ICMP) traffic destined for the Routing Engine by specifying a list of prefix sources that contain allowed BGP peers.

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you create a stateless firewall filter that blocks all TCP connection attempts to port 179 from all requesters except BGP peers that have a specified prefix.

A source prefix list, **plist_bgp179**, is created that specifies the list of source prefixes that contain allowed BGP peers.

The stateless firewall filter **filter_bgp179** matches all packets from the source prefix list **plist_bgp179** to the destination port number 179.

Configuration

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*.

- Configure the Filter on page 312
- Results on page 313

CLI Quick Configuration

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

```
set policy-options prefix-list plist_bgp179 apply-path "protocols bgp group <*> neighbor
  <*>"
set firewall family inet filter filter_bgp179 term 1 from source-address 0.0.0.0/0
set firewall family inet filter filter_bgp179 term 1 from source-prefix-list plist_bgp179 except
set firewall family inet filter filter_bgp179 term 1 from destination-port bgp
set firewall family inet filter filter_bgp179 term 1 then reject
set firewall family inet filter filter_bgp179 term 2 then accept
set interfaces lo0 unit 0 family inet filter input filter_bgp179
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

Configure the Filter

Step-by-Step Procedure

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

To configure the filter:

1. Expand the prefix list **bgp179** to include all prefixes pointed to by the BGP peer group defined by **protocols bgp group <*> neighbor <*>**.

   ```
   [edit policy-options prefix-list plist_bgp179]
   user@host# set apply-path "protocols bgp group <*> neighbor <*>"
   ```

2. Define the filter term that rejects TCP connection attempts to port 179 from all requesters except the specified BGP peers.

   ```
   [edit firewall family inet filter filter_bgp179]
   user@host# set term term1 from source-address 0.0.0.0/0
   user@host# set term term1 from source-prefix-list bgp179 except
   ```

```
user@host# set term term1 from destination-port bgp
user@host# set term term1 then reject
```

3. Define the other filter term to accept all packets.

```
[edit firewall family inet filter filter_bgp179]
user@host# set term term2 then accept
```

4. Apply the firewall filter to the loopback interface.

```
[edit interfaces lo0 unit 0 family inet]
user@host# set filter input filter_bgp179
user@host# set address 127.0.0.1/32
```

Results

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

```
user@host# show firewall
family inet {
    filter filter_bgp179 {
        term 1 {
            from {
                source-address {
                    0.0.0.0/0;
                }
                source-prefix-list {
                    plist_bgp179 except;
                }
                destination-port bgp;
            }
            then {
                reject;
            }
        }
        term 2 {
            then {
                accept;
            }
        }
    }
}
user@host# show interfaces
lo0 {
    unit 0 {
        family inet {
            filter {
                input filter_bgp179;
            }
            address 127.0.0.1/32;
        }
    }
}
```

```
user@host# show policy-options
prefix-list plist_bgp179 {
  apply-path "protocols bgp group <*> neighbor <*>";
}
```

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

Repeat the procedure, where appropriate, for every BGP-enabled device in the network, using the appropriate interface names and addresses for each BGP-enabled device.

Verification

Confirm that the configuration is working properly.

Displaying the Firewall Filter Applied to the Loopback Interface

Purpose Verify that the firewall filter **filter_bgp179** is applied to the IPv4 input traffic at logical interface **lo0.0**.

Action Use the **show interfaces statistics** operational mode command for logical interface **lo0.0**, and include the **detail** option. Under the **Protocol inet** section of the command output section, the **Input Filters** field displays the name of the stateless firewall filter applied to the logical interface in the input direction:

```
[edit]
user@host> show interfaces statistics lo0.0 detail
  Logical interface lo0.0 (Index 321) (SNMP ifIndex 16) (Generation 130)
    Flags: SNMP-Traps Encapsulation: Unspecified
    Traffic statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0
    Local statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0
    Transit statistics:
     Input  bytes  :                   0                   0 bps
     Output bytes  :                   0                   0 bps
     Input  packets:                   0                   0 pps
     Output packets:                   0                   0 pps
    Protocol inet, MTU: Unlimited, Generation: 145, Route table: 0
      Flags: Sendbcast-pkt-to-re
      Input Filters: filter_bgp179
      Addresses, Flags: Primary
        Destination: Unspecified, Local: 127.0.0.1, Broadcast: Unspecified,
Generation: 138
```

Example: Limiting TCP Segment Size for BGP

This example shows how to avoid Internet Control Message Protocol (ICMP) vulnerability issues by limiting TCP segment size when you are using maximum transmission unit

(MTU) discovery. Using MTU discovery on TCP paths is one method of avoiding BGP packet fragmentation.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

TCP negotiates a maximum segment size (MSS) value during session connection establishment between two peers. The MSS value negotiated is primarily based on the maximum transmission unit (MTU) of the interfaces to which the communicating peers are directly connected. However, due to variations in link MTU on the path taken by the TCP packets, some packets in the network that are well within the MSS value might be fragmented when the packet size exceeds the link's MTU.

To configure the TCP MSS value, include the tcp-mss statement with a segment size from 1 through 4096.

If the router receives a TCP packet with the SYN bit and the MSS option set, and the MSS option specified in the packet is larger than the MSS value specified by the **tcp-mss** statement, the router replaces the MSS value in the packet with the lower value specified by the **tcp-mss** statement.

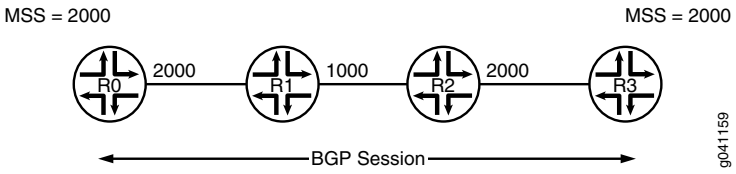
The configured MSS value is used as the maximum segment size for the sender. The assumption is that the TCP MSS value used by the sender to communicate with the BGP neighbor is the same as the TCP MSS value that the sender can accept from the BGP neighbor. If the MSS value from the BGP neighbor is less than the MSS value configured, the MSS value from the BGP neighbor is used as the maximum segment size for the sender.

This feature is supported with TCP over IPv4 and TCP over IPv6.

Topology Diagram

Figure 34 on page 315 shows the topology used in this example.

Figure 34: TCP Maximum Segment Size for BGP



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.

R0

```
set interfaces fe-1/2/0 unit 1 family inet address 1.1.0.1/30
set interfaces lo0 unit 1 family inet address 10.255.14.179/32
set protocols bgp group-int tcp-mss 2020
set protocols bgp group int type internal
set protocols bgp group int local-address 10.255.14.179
set protocols bgp group int mtu-discovery
set protocols bgp group int neighbor 10.255.71.24 tcp-mss 2000
set protocols bgp group int neighbor 10.255.14.177
set protocols bgp group int neighbor 10.0.14.4 tcp-mss 4000
set protocols ospf area 0.0.0.0 interface fe-1/2/0.1
set protocols ospf area 0.0.0.0 interface 10.255.14.179
set routing-options autonomous-system 65000
```

Step-by-Step Procedure

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

To configure Router R0:

1. Configure the interfaces.

   ```
   [edit interfaces]
   user@R0# set fe-1/2/0 unit 1 family inet address 1.1.0.1/30
   user@R0# set lo0 unit 1 family inet address 10.255.14.179/32
   ```

2. Configure an interior gateway protocol (IGP), OSPF in this example.

   ```
   [edit protocols ospf area 0.0.0.0]
   user@R0# set interface fe-1/2/0.1
   user@R0# set interface 10.255.14.179
   ```

3. Configure one or more BGP groups.

   ```
   [edit protocols bgp group int]
   user@R0# set type internal
   user@R0# set local-address 10.255.14.179
   ```

4. Configure MTU discovery to prevent packet fragmentation.

   ```
   [edit protocols bgp group int]
   user@R0# set mtu-discovery
   ```

5. Configure the BGP neighbors, with the TCP MSS set globally for the group or specifically for the various neighbors.

   ```
   [edit protocols bgo group int]
   user@R0# set tcp-mss 2020
   user@R0# set neighbor 10.255.14.177
   user@R0# set neighbor 10.255.71.24 tcp-mss 2000
   user@R0# set neighbor 10.0.14.4 tcp-mss 4000
   ```

NOTE: The TCP MSS neighbor setting overrides the group setting.

6. Configure the local autonomous system.

```
[edit routing-options]
user@R0# set autonomous-system 65000
```

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

```
user@R0# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 1.1.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 10.255.14.179/32;
    }
  }
}
```

```
user@R0# show protocols
bgp {
  group int {
    type internal;
    local-address 10.255.14.179;
    mtu-discovery;
    tcp-mss 2020;
    neighbor 10.255.71.24 {
      tcp-mss 2000;
    }
    neighbor 10.255.14.177;
    neighbor 10.0.14.4 {
      tcp-mss 4000;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface fe-1/2/0.1;
    interface 10.255.14.179;
  }
}
```

```
user@R0# show routing-options
autonomous-system 65000;
```

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

Verification

To confirm that the configuration is working properly, run the following commands:

- **show system connections extensive | find <neighbor-address>**, to check the negotiated TCP MSS value.
- **monitor traffic interface**, to monitor BGP traffic and to make sure that the configured TCP MSS value is used as the MSS option in the TCP SYN packet.

Troubleshooting

- MSS Calculation with MTU Discovery on page 318

MSS Calculation with MTU Discovery

Problem Consider an example in which two routing devices (R1 and R2) have an internal BGP (IBGP) connection. On both of the routers, the connected interfaces have 4034 as the IPv4 MTU.

```
user@R1# show protocols bgp | display set
[edit]
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 45.45.45.2
set protocols bgp group ibgp mtu-discovery
set protocols bgp group ibgp neighbor 45.45.45.1

user@R1# run show interfaces xe-0/0/3 extensive | match mtu

  Link-level type: Ethernet, MTU: 4048, LAN-PHY mode, Speed: 10Gbps,
    FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
    Protocol inet, MTU: 4034, Generation: 180, Route table: 0
    Protocol multiservice, MTU: Unlimited, Generation: 181, Route table: 0
```

In the following packet capture on Device R1, the negotiated MSS is 3994. In the **show system connections extensive** information for MSS, it is set to 2048.

```
05:50:01.575218 Out
        Juniper PCAP Flags [Ext], PCAP Extension(s) total length 16
          Device Media Type Extension TLV #3, length 1, value: Ethernet (1)
          Logical Interface Encapsulation Extension TLV #6, length 1, value:
Ethernet (14)
          Device Interface Index Extension TLV #1, length 2, value: 137
          Logical Interface Index Extension TLV #4, length 4, value: 69
        -----original packet-----
        00:21:59:e1:e8:03 > 00:19:e2:20:79:01, ethertype IPv4 (0x0800), length
78: (tos 0xc0, ttl 64, id 53193, offset 0, flags [DF], proto: TCP (6), length:
64) 45.45.45.2.62840 > 45.45.45.1.bgp: S 2939345813:2939345813(0) win 16384 **mss
 3994,nop,wscale 0,nop,nop,timestamp 70559970 0,sackOK,eol>
05:50:01.575875 In
        Juniper PCAP Flags [Ext, no-L2, In], PCAP Extension(s) total length 16
          Device Media Type Extension TLV #3, length 1, value: Ethernet (1)
          Logical Interface Encapsulation Extension TLV #6, length 1, value:
Ethernet (14)
          Device Interface Index Extension TLV #1, length 2, value: 137
          Logical Interface Index Extension TLV #4, length 4, value: 69
```

```
       -----original packet-----
  PFE proto 2 (ipv4): (tos 0xc0, ttl 255, id 37709, offset 0, flags [DF], proto:
 TCP (6), length: 64) 45.45.45.1.bgp > 45.45.45.2.62840: S 2634967984:2634967984(0)
 ack 2939345814 win 16384 **mss 3994,nop,wscale 0,nop,nop,timestamp 174167273
70559970,sackOK,eol>

user@R1# run show system connections extensive | find 45.45

tcp4       0      0  45.45.45.2.62840                         45.45.45.1.179
                                ESTABLISHED
   sndsbcc:          0 sndsbmbcnt:          0  sndsbmbmax:     131072
sndsblowat:       2048 sndsbhiwat:      16384
   rcvsbcc:          0 rcvsbmbcnt:          0  rcvsbmbmax:     131072
rcvsblowat:          1 rcvsbhiwat:      16384
   proc id:      19725  proc name:        rpd
       iss: 2939345813      sndup: 2939345972
    snduna: 2939345991     sndnxt: 2939345991      sndwnd:      16384
    sndmax: 2939345991    sndcwnd:      10240 sndssthresh: 1073725440
       irs: 2634967984      rcvup: 2634968162
    rcvnxt: 2634968162     rcvadv: 2634984546      rcvwnd:      16384
       rtt:          0       srtt:       1538        rttv:       1040
    rxtcur:       1200   rxtshift:          0       rtseq: 2939345972
    rttmin:       1000  mss:        2048
```

Solution This is expected behavior with Junos OS. The MSS value is equal to the MTU value minus the IP or IPv6 and TCP headers. This means that the MSS value is generally 40 bytes less than the MTU (for IPv4) and 60 bytes less than the MTU (for IPv6). This value is negotiated between the peers. In this example, it is 4034 - 40 = 3994. Junos OS then rounds this value to a multiple of 2 KB. The value is 3994 / 2048 * 2048=2048. So it is not necessary to see same MSS value with in the **show system connections** output.

3994 / 2048 = 1.95

1.95 is rounded to 1.

1 * 2048 = 2048

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

CHAPTER 9

BGP Flap Configuration

Example: Preventing BGP Session Resets

Understanding BGP Session Resets

Certain configuration actions and events cause BGP sessions to be reset (dropped and then reestablished).

If you configure both route reflection and VPNs on the same routing device, the following modifications to the route reflection configuration cause current BGP sessions to be reset:

- Adding a cluster ID—If a BGP session shares the same autonomous system (AS) number with the group where you add the cluster ID, all BGP sessions are reset regardless of whether the BGP sessions are contained in the same group.
- Creating a new route reflector—If you have an internal BGP (IBGP) group with an AS number and create a new route reflector group with the same AS number, all BGP sessions in the IBGP group and the new route reflector group are reset.
- Changing configuration statements that affect BGP peers, such as renaming a BGP group, resets the BGP sessions.
- If you change the address family specified in the **[edit protocols bgp family]** hierarchy level, all current BGP sessions on the routing device are dropped and then reestablished.

Example: Preventing BGP Session Flaps When VPN Families Are Configured

This example shows a workaround for a known issue in which BGP sessions sometimes go down and then come back up (in other words, flap) when virtual private network (VPN) families are configured. If any VPN family (for example, **inet-vpn**, **inet6-vpn**, **inet-mpvn**, **inet-mdt**, **inet6-mpvn**, **l2vpn**, **iso-vpn**, and so on) is configured on a BGP master instance, a flap of either a route reflector (RR) internal BGP (IBGP) session or an external

BGP (EBGP) session causes flaps of other BGP sessions configured with the same VPN family.

Requirements

Before you begin:

- Configure router interfaces.
- Configure an interior gateway protocol (IGP).
- Configure BGP.
- Configure VPNs.

Overview

When a router or switch is configured as either a route reflector (RR) or an AS boundary router (an external BGP peer) and a VPN family (for example, the **family inet-vpn unicast** statement) is configured, a flap of either the RR IBGP session or the EBGP session causes flaps of all other BGP sessions that are configured with the **family inet-vpn unicast** statement. This example shows how to prevent these unnecessary session flaps.

The reason for the flapping behavior is related to BGP operation in Junos OS when originating VPN routes.

BGP has the following two modes of operation with respect to originating VPN routes:

- If BGP does not need to propagate VPN routes because the session has no EBGP peer and no RR clients, BGP exports VPN routes directly from the ***instance*.inet.0** routing table to other PE routers. This behavior is efficient in that it avoids the creation of two copies of many routes (one in the ***instance*.inet.0** table and one in the **bgp.l3vpn.0** table).
- If BGP does need to propagate VPN routes because the session has an EBGP peer or RR clients, BGP first exports the VPN routes from the ***instance*.inet.0** table to the **bgp.l3vpn.0** table. Then BGP exports the routes to other PE routers. In this scenario, two copies of the route are needed to enable best-route selection. A PE router might receive the same VPN route from a CE device and also from an RR client or EBGP peer.

When, because of a configuration change, BGP transitions from needing two copies of a route to not needing two copies of a route (or the reverse), all sessions over which VPN routes are exchanged go down and then come back up. Although this example focuses on the **family inet-vpn unicast** statement, the concept applies to all VPN network layer reachability information (NLRI) families. This issue impacts logical systems as well. All BGP sessions in the master instance related to the VPN NLRI family are brought down to implement the table advertisement change for the VPN NLRI family. Changing an RR to a non-RR or the reverse (by adding or removing the **cluster** statement) causes the table advertisement change. Also, configuring the first EBGP session or removing the EBGP session from the configuration in the master instance for a VPN NLRI family causes the table advertisement change.

The way to prevent these unnecessary session flaps is to configure an extra RR client or EBGP session as a passive session with a neighbor address that does not exist. This example focuses on the EBGP case, but the same workaround works for the RR case.

When a session is passive, the routing device does not send Open requests to a peer. Once you configure the routing device to be passive, the routing device does not originate the TCP connection. However, when the routing device receives a connection from the peer and an Open message, it replies with another BGP Open message. Each routing device declares its own capabilities.

Figure 35 on page 324 shows the topology for the EBGP case. Router R1 has an IBGP session with Routers R2 and R3 and an EBGP session with Router R4. All sessions have the **family inet-vpn unicast** statement configured. If the R1-R4 EBGP session flaps, the R1-R2 and R1-R3 BGP sessions flap also.

Figure 35: Topology for the EBGP Case

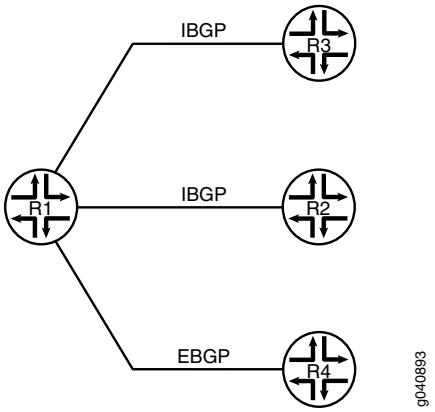
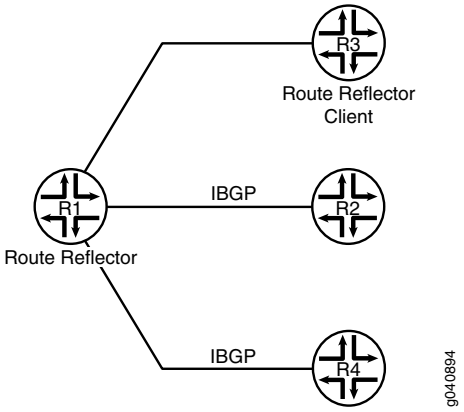


Figure 36 on page 324 shows the topology for the RR case. Router R1 is the RR, and Router R3 is the client. Router R1 has IBGP sessions with Routers R2 and R3. All sessions have the **family inet-vpn unicast** statement configured. If the R1-R3 session flaps, the R1-R2 and R1-R4 sessions flap also.

Figure 36: Topology for the RR Case



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 protocols bgp family inet-vpn unicast
set protocols bgp family l2vpn signaling
set protocols bgp group R1-R4 type external
set protocols bgp group R1-R4 local-address 4.4.4.2
set protocols bgp group R1-R4 neighbor 4.4.4.1 peer-as 200
set protocols bgp group R1-R2-R3 type internal
set protocols bgp group R1-R2-R3 log-updown
set protocols bgp group R1-R2-R3 local-address 15.15.15.15
set protocols bgp group R1-R2-R3 neighbor 12.12.12.12
set protocols bgp group R1-R2-R3 neighbor 13.13.13.13
```

```
set protocols bgp group Fake type external
set protocols bgp group Fake passive
set protocols bgp group Fake neighbor 100.100.100.100 peer-as 500
```

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 EBGP scenario:

1. Configure one or more VPN families.

   ```
   [edit protocols bgp]
   user@R1# set family inet-vpn unicast
   user@R1# set family l2vpn signaling
   ```

2. Configure the EBGP session.

   ```
   [edit protocols bgp]
   user@R1# set group R1-R4 type external
   user@R1# set group R1-R4 local-address 4.4.4.2
   user@R1# set group R1-R4 neighbor 4.4.4.1 peer-as 200
   ```

3. Configure the IBGP sessions.

   ```
   [edit protocols bgp]
   user@R1# set group R1-R2-R3 type internal
   user@R1# set group R1-R2-R3 local-address 15.15.15.15
   user@R1# set group R1-R2-R3 neighbor 12.12.12.12
   user@R1# set group R1-R2-R3 neighbor 13.13.13.13
   ```

4. (Optional) Configure BGP so that it generates a **syslog** message whenever a BGP peer makes a state transition.

   ```
   [edit protocols bgp]
   user@R1# set group R1-R2-R3 log-updown
   ```

   Enabling the **log-updown** statement causes BGP state transitions to be logged at **warning** level.

Step-by-Step Procedure

To verify that unnecessary session flaps are occurring:

1. Run the **show bgp summary** command to verify that the sessions have been established.

   ```
   user@R1> show bgp summary
   Groups: 2 Peers: 3 Down peers: 0
   Table       Tot Paths Act Paths Suppressed History Damp State Pending
   bgp.l3vpn.0 0         0         0          0       0          0
   bgp.l2vpn.0 0         0         0          0       0          0
   inet.0      0         0         0          0       0          0
   Peer        AS  InPkt OutPkt OutQ Flaps Last Up/Dwn
   State|#Active/Received/Accepted/Damped...
   4.4.4.1     200 6     5      0    0     1:08 Establ
   bgp.l3vpn.0: 0/0/0/0
   bgp.l2vpn.0: 0/0/0/0
   12.12.12.12 100 3     7      0    0     1:18 Establ
   bgp.l3vpn.0: 0/0/0/0
   bgp.l2vpn.0: 0/0/0/0
   ```

```
13.13.13.13 100 3     6      0    0     1:14 Establ
bgp.l3vpn.0: 0/0/0/0
bgp.l2vpn.0: 0/0/0/0
```

2. Deactivate the EBGP session.

```
user@R1# deactivate group R1-R4
user@R1# commit
```

```
Mar 10 18:27:40 R1: rpd[1464]: bgp_peer_delete:6589: NOTIFICATION sent to 4.4.4.1 (External AS 200): code
 6 (Cease) subcode 3 (Peer Unconfigured), Reason: Peer Deletion
Mar 10 18:27:40 R1: rpd[1464]: bgp_adv_main_update:7253: NOTIFICATION sent to 12.12.12.12 (Internal AS
100): code 6 (Cease) subcode 6 (Other Configuration Change), Reason: Configuration change - VPN table
advertise
Mar 10 18:27:40 R1: rpd[1464]: bgp_adv_main_update:7253: NOTIFICATION sent to 13.13.13.13 (Internal AS
100): code 6 (Cease) subcode 6 (Other Configuration Change), Reason: Configuration change - VPN table
advertise
```

3. Run the **show bgp summary** command to view the session flaps.

```
user@R1> show bgp summary
Groups: 1 Peers: 2 Down peers: 2
Table       Tot Paths Act Paths Suppressed History Damp State Pending
bgp.l3vpn.0 0         0         0          0       0          0
bgp.l2vpn.0 0         0         0          0       0          0
inet.0      0         0         0          0       0          0
Peer        AS   InPkt  OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
12.12.12.12 100  4      9      0    1     19   Active
13.13.13.13 100  4      8      0    1     19   Active
```

```
user@R1> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table       Tot Paths Act Paths Suppressed History Damp State Pending
bgp.l3vpn.0 0         0         0          0       0          0
bgp.l2vpn.0 0         0         0          0       0          0
inet.0      0         0         0          0       0          0
Peer        AS  InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
12.12.12.12 100 2     3      0    1     0    Establ
bgp.l3vpn.0: 0/0/0/0
bgp.l2vpn.0: 0/0/0/0
13.13.13.13 100 2     3      0    1     0    Establ
bgp.l3vpn.0: 0/0/0/0
bgp.l2vpn.0: 0/0/0/0
```

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 prevent unnecessary BGP session flaps:

1. Add a passive EBGP session with a neighbor address that does not exist in the peer autonomous system (AS).

```
[edit protocols bgp]
user@R1# set group Fake type external
user@R1# set group Fake passive
user@R1# set neighbor 100.100.100.100 peer-as 500
```

2. Run the **show bgp summary** command to verify that the real sessions have been established and the passive session is idle.

```
user@R1> show bgp summary
Groups: 3 Peers: 4 Down peers: 1
Table       Tot Paths Act Paths Suppressed History Damp State Pending
bgp.l3vpn.0 0         0         0          0       0          0
bgp.l2vpn.0 0         0         0          0       0          0
Peer            AS  InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
4.4.4.1         200 9500  9439   0    0     2d   23:14:23 Establ
bgp.l3vpn.0: 0/0/0/0
bgp.l2vpn.0: 0/0/0/0
12.12.12.12     100 10309 10239  0    0     3d    5:17:49 Establ
bgp.l3vpn.0: 0/0/0/0
13.13.13.13     100 10306 10241  0    0     3d    5:18:25 Establ
bgp.l3vpn.0: 0/0/0/0
100.100.100.100 500 0     0      0    0     2d   23:38:52 Idle
```

Verification

Confirm that the configuration is working properly.

- Bringing Down the EBGP Session on page 327
- Verifying That the IBGP Sessions Remain Up on page 327

Bringing Down the EBGP Session

Purpose Try to cause the flap issue after the workaround is configured.

Action

```
user@R1# deactivate group R1-R4
user@R1# commit
```

Verifying That the IBGP Sessions Remain Up

Purpose Make sure that the IBGP sessions do not flap after the EBGP session is deactivated.

Action

```
user@R1> show bgp summary
Groups: 2 Peers: 3 Down peers: 1
Table       Tot Paths Act Paths Suppressed History Damp State Pending
bgp.l3vpn.0 0         0         0          0       0          0
bgp.l2vpn.0 0         0         0          0       0          0
Peer            AS  InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
12.12.12.12     100 10312 10242  0    0     3d    5:19:01  Establ
bgp.l3vpn.0: 0/0/0/0
13.13.13.13     100 10309 10244  0    0     3d    5:19:37  Establ
bgp.l3vpn.0: 0/0/0/0
100.100.100.100 500 0     0      0    0     2d   23:40:04  Idle

user@R1> show bgp summary
Groups: 3 Peers: 4 Down peers: 1
Table       Tot Paths Act Paths Suppressed History Damp State Pending
bgp.l3vpn.0 0         0         0          0       0          0
bgp.l2vpn.0 0         0         0          0       0          0
Peer            AS  InPkt OutPkt OutQ Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
4.4.4.1         200 5     4      0    0     28             Establ
bgp.l3vpn.0: 0/0/0/0
bgp.l2vpn.0: 0/0/0/0
12.12.12.12     100 10314 10244  0    0     3d    5:19:55  Establ
bgp.l3vpn.0: 0/0/0/0
13.13.13.13     100 10311 10246  0    0     3d    5:20:31  Establ
bgp.l3vpn.0: 0/0/0/0
100.100.100.100 500 0     0      0    0     2d   23:40:58  Idle
```

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

Examples: Configuring BGP Flap Damping

Understanding Damping Parameters

BGP *route flapping* describes the situation in which BGP systems send an excessive number of update messages to advertise network reachability information. BGP *flap damping* is a method of reducing the number of update messages sent between BGP peers, thereby reducing the load on these peers, without adversely affecting the route convergence time for stable routes.

Flap damping reduces the number of update messages by marking routes as ineligible for selection as the active or preferable route. Marking routes in this way leads to some delay, or *suppression*, in the propagation of route information, but the result is increased network stability. You typically apply flap damping to external BGP (EBGP) routes (routes in different ASs). You can also apply flap damping within a confederation, between

confederation member ASs. Because routing consistency within an AS is important, do not apply flap damping to internal BGP (IBGP) routes. (If you do, it is ignored.) The exception to this rule is when flap damping is applied at the address family level, which is supported in Junos OS Release 12.2 and later. When you apply flap damping at the address family level, it works for both IBGP and EBGP.

By default, route flap damping is not enabled. Damping is applied to external peers and to peers at confederation boundaries.

When you enable damping, default parameters are applied, as summarized in Table 5 on page 329.

Table 5: Damping Parameters

Damping Parameter	Description	Default Value	Possible Values
half-life ***minutes***	Decay half-life—Number of minutes after which an arbitrary value is halved if a route stays stable.	15 (minutes)	1 through 45
max-suppress ***minutes***	Maximum hold-down time for a route, in minutes.	60 (minutes)	1 through 720
reuse	Reuse threshold—Arbitrary value below which a suppressed route can be used again.	750	1 through 20,000
suppress	Cutoff (suppression) threshold—Arbitrary value above which a route can no longer be used or included in advertisements.	3000	1 through 20,000

To change the default BGP flap damping values, you define actions by creating a named set of damping parameters and including it in a routing policy with the damping action. For the damping routing policy to work, you also must enable BGP route flap damping.

Example: Configuring Damping Parameters

This example shows how to configure damping parameters.

Requirements

Before you begin, configure router interfaces and configure routing protocols.

Overview

This example has three routing devices. Device R2 has external BGP (EBGP) connections with Device R1 and Device R3.

Device R1 and Device R3 have some static routes configured for testing purposes, and these static routes are advertised through BGP to Device R2.

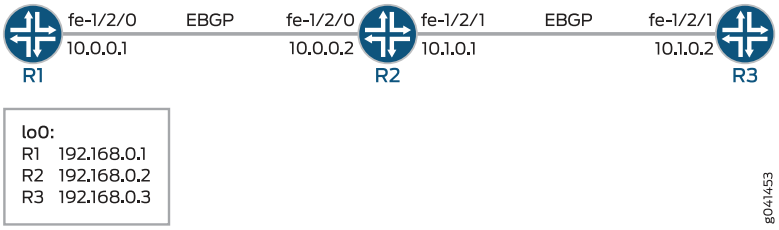
Device R2 damps routes received from Device R1 and Device R3 according to these criteria:

- Damp all prefixes with a mask length equal to or greater than 17 more aggressively than routes with a mask length between 9 and 16.
- Damp routes with a mask length between 0 and 8, inclusive, less than routes with a mask length greater than 8.
- Do not damp the 10.128.0.0/9 prefix at all.

The routing policy is evaluated when routes are being exported from the routing table into the forwarding table. Only the active routes are exported from the routing table.

Figure 37 on page 330 shows the sample network.

Figure 37: BGP Flap Damping Topology



"CLI Quick Configuration" on page 330 shows the configuration for all of the devices in Figure 37 on page 330.

The section "Step-by-Step Procedure" on page 331 describes the steps on Device R2.

Configuration

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct-and-static
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.0.0.2
set policy-options policy-statement send-direct-and-static term 1 from protocol direct
set policy-options policy-statement send-direct-and-static term 1 from protocol static
set policy-options policy-statement send-direct-and-static term 1 then accept
set routing-options static route 172.16.0.0/16 reject
set routing-options static route 172.16.128.0/17 reject
set routing-options static route 172.16.192.0/20 reject
set routing-options static route 10.0.0.0/9 reject
set routing-options static route 224.0.0.0/7 reject
set routing-options static route 10.224.0.0/11 reject
set routing-options static route 0.0.0.0/0 reject
set routing-options autonomous-system 100
```

Device R2

```
set interfaces fe-1/2/0 unit 0 family inet address 10.0.0.2/30
```

```
set interfaces fe-1/2/1 unit 0 family inet address 10.1.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols bgp damping
set protocols bgp group ext type external
set protocols bgp group ext import damp
set protocols bgp group ext export send-direct
set protocols bgp group ext neighbor 10.0.0.1 peer-as 100
set protocols bgp group ext neighbor 10.1.0.2 peer-as 300
set policy-options policy-statement damp term 1 from route-filter 10.128.0.0/9 exact
  damping dry
set policy-options policy-statement damp term 1 from route-filter 0.0.0.0/0
  prefix-length-range /0-/8 damping timid
set policy-options policy-statement damp term 1 from route-filter 0.0.0.0/0
  prefix-length-range /17-/32 damping aggressive
set policy-options policy-statement send-direct term 1 from protocol direct
set policy-options policy-statement send-direct term 1 then accept
set policy-options damping aggressive half-life 30
set policy-options damping aggressive suppress 2500
set policy-options damping timid half-life 5
set policy-options damping dry disable
set routing-options autonomous-system 200
```

Device R3

```
set interfaces fe-1/2/1 unit 0 family inet address 10.1.0.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set protocols bgp group ext type external
set protocols bgp group ext export send-direct-and-static
set protocols bgp group ext peer-as 200
set protocols bgp group ext neighbor 10.1.0.1
set policy-options policy-statement send-direct-and-static term 1 from protocol direct
set policy-options policy-statement send-direct-and-static term 1 from protocol static
set policy-options policy-statement send-direct-and-static term 1 then accept
set routing-options static route 10.128.0.0/9 reject
set routing-options autonomous-system 300
```

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 damping parameters:

1. Configure the interfaces.

```
[edit interfaces]
user@R2# set fe-1/2/0 unit 0 family inet address 10.0.0.2/30

user@R2# set fe-1/2/1 unit 0 family inet address 10.1.0.1/30

user@R2# set lo0 unit 0 family inet address 192.168.0.2/32
```

2. Configure the BGP neighbors.

```
[edit protocols bgp group ext]
user@R2# set type external
user@R2# set neighbor 10.0.0.1 peer-as 100
user@R2# set neighbor 10.1.0.2 peer-as 300
```

3. Create and configure the damping parameter groups.

```
[edit policy-options]
user@R2# set damping aggressive half-life 30
user@R2# set damping aggressive suppress 2500
user@R2# set damping timid half-life 5
user@R2# set damping dry disable
```

4. Configure the damping policy.

```
[edit policy-options policy-statement damp term 1]
user@R2# set from route-filter 10.128.0.0/9 exact damping dry
user@R2# set from route-filter 0.0.0.0/0 prefix-length-range /0-/8 damping timid
user@R2# set from route-filter 0.0.0.0/0 prefix-length-range /17-/32 damping
  aggressive
```

5. Enable damping for BGP.

```
[edit protocols bgp]
user@R2# set damping
```

6. Apply the policy as an import policy for the BGP neighbor.

```
[edit protocols bgp group ext]
user@R2# set import damp
```

NOTE: You can refer to the same routing policy one or more times in the same or different import statements.

7. Configure an export policy.

```
[edit policy-options policy-statement send-direct term 1]
user@R2# set from protocol direct
user@R2# set then accept
```

8. Apply the export policy.

```
[edit protocols bgp group ext]
user@R2# set export send-direct
```

9. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@R2# set autonomous-system 200
```

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

```
user@R2# show interfaces
fe-1/2/0 {
  unit 0 {
    family inet {
      address 10.0.0.2/30;
    }
```

```
    }
  }
  fe-1/2/1 {
    unit 0 {
      family inet {
        address 10.1.0.1/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 192.168.0.2/32;
      }
    }
  }
```

```
user@R2# show protocols
bgp {
  damping;
  group ext {
    type external;
    import damp;
    export send-direct;
    neighbor 10.0.0.1 {
      peer-as 100;
    }
    neighbor 10.1.0.2 {
      peer-as 300;
    }
  }
}
```

```
user@R2# show policy-options
policy-statement damp {
  term 1 {
    from {
      route-filter 10.128.0.0/9 exact damping dry;
      route-filter 0.0.0.0/0 prefix-length-range /0-/8 damping timid;
      route-filter 0.0.0.0/0 prefix-length-range /17-/32 damping aggressive;
    }
  }
}
policy-statement send-direct {
  term 1 {
    from protocol direct;
    then accept;
  }
}
damping aggressive {
  half-life 30;
  suppress 2500;
}
damping timid {
  half-life 5;
}
damping dry {
```

```
    disable;
}

user@R2# show routing-options
autonomous-system 200;
```

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

Verification

Confirm that the configuration is working properly.

Causing Some Routes to Flap

Purpose To verify your route flap damping policy, some routes must flap. Having a live Internet feed almost guarantees that a certain number of route flaps will be present. If you have control over a remote system that is advertising the routes, you can modify the advertising router's policy to effect the advertisement and withdrawal of all routes or of a given prefix. In a test environment, you can cause routes to flap by clearing the BGP neighbors or by restarting the routing process on the BGP neighbors, as shown here.

Action From operational mode on Device R1 and Device R3, enter the **restart routing** command.

CAUTION: Use this command cautiously in a production network.

```
user@R1> restart routing

R1 started, pid 10474

user@R3> restart routing

R3 started, pid 10478
```

Meaning On Device R2, all of the routes from the neighbors are withdrawn and re-advertised.

Checking the Route Flaps

Purpose View the number of neighbor flaps.

Action From operational mode, enter the **show bgp summary** command.

```
user@R2> show bgp summary
```

```
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                      12          1         11          0         11          0
Peer                    AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
10.0.0.1               100         10         10       0       4        2:50
0/9/0/9              0/0/0/0
10.1.0.2               300         10         10       0       4        2:53
1/3/1/2              0/0/0/0
```

Meaning This output was captured after the routing process was restarted on Device R2's neighbors four times.

Verifying Route Flap Damping

Purpose Verify that routes are being hidden due to damping.

Action From operational mode, enter the **show route damping suppressed** command.

```
user@R2> show route damping suppressed

inet.0: 15 destinations, 17 routes (6 active, 0 holddown, 11 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0           [BGP ] 00:00:12, localpref 100
                      AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
10.0.0.0/9          [BGP ] 00:00:12, localpref 100
                      AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
10.0.0.0/30         [BGP ] 00:00:12, localpref 100
                      AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
10.1.0.0/30         [BGP ] 00:00:15, localpref 100
                      AS path: 300 I, validation-state: unverified
                    > to 10.1.0.2 via fe-1/2/1.0
10.224.0.0/11       [BGP ] 00:00:12, localpref 100
                      AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
172.16.0.0/16       [BGP ] 00:00:12, localpref 100
                      AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
172.16.128.0/17     [BGP ] 00:00:12, localpref 100
                      AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
172.16.192.0/20     [BGP ] 00:00:12, localpref 100
                      AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
192.168.0.1/32      [BGP ] 00:00:12, localpref 100
                      AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
192.168.0.3/32      [BGP ] 00:00:15, localpref 100
                      AS path: 300 I, validation-state: unverified
                    > to 10.1.0.2 via fe-1/2/1.0
224.0.0.0/7         [BGP ] 00:00:12, localpref 100
                      AS path: 100 I, validation-state: unverified
                    > to 10.0.0.1 via fe-1/2/0.0
```

Meaning The output shows some routing instability. Eleven routes are hidden due to damping.

Displaying the Details of a Damped Route

Purpose Display the details of damped routes.

Action From operational mode, enter the **show route damping suppressed 172.16.192.0/20 detail** command.

```
user@R2> show route damping suppressed 172.16.192.0/20 detail

inet.0: 15 destinations, 17 routes (6 active, 0 holddown, 11 hidden)
172.16.192.0/20 (1 entry, 0 announced)
         BGP                /-101
                Next hop type: Router, Next hop index: 758
                Address: 0x9414484
                Next-hop reference count: 9
                Source: 10.0.0.1
                Next hop: 10.0.0.1 via fe-1/2/0.0, selected
                Session Id: 0x100201
                State: <Hidden Ext>
                Local AS:   200 Peer AS:   100
                Age: 52
                Validation State: unverified
                Task: BGP_100.10.0.0.1+55922
                AS path: 100 I
                Localpref: 100
                Router ID: 192.168.0.1
                Merit (last update/now): 4278/4196
                damping-parameters: aggressive
                Last update:       00:00:52 First update:       01:01:55
                Flaps: 8
                Suppressed. Reusable in:       01:14:40
                Preference will be: 170
```

Meaning This output indicates that the displayed route has a mask length that is equal to or greater than /17, and confirms that it has been correctly mapped to the aggressive damping profile. You can also see the route's current (and last) figure of merit value, and when the route is expected to become active if it remains stable.

Verifying That Default Damping Parameters Are in Effect

Purpose Locating a damped route with a /16 mask confirms that the default parameters are in effect.

Action From operational mode, enter the **show route damping suppressed detail | match 0/16** command.

```
user@R2> show route damping suppressed detail | match 0/16

172.16.0.0/16 (1 entry, 0 announced)

user@R2> show route damping suppressed 172.16.0.0/16 detail

inet.0: 15 destinations, 17 routes (6 active, 0 holddown, 11 hidden)
172.16.0.0/16 (1 entry, 0 announced)
```



```
BGP                 /-101
       Next hop type: Router, Next hop index: 758
       Address: 0x9414484
       Next-hop reference count: 9
       Source: 10.0.0.1
       Next hop: 10.0.0.1 via fe-1/2/0.0, selected
       Session Id: 0x100201
       State: <Hidden Ext>
       Local AS:   200 Peer AS:   100
       Age: 1:58
       Validation State: unverified
       Task: BGP_100.10.0.0.1+55922
       AS path: 100 I
       Localpref: 100
       Router ID: 192.168.0.1
       Merit (last update/now): 3486/3202
       Default damping parameters used
       Last update:       00:01:58 First update:       01:03:01
       Flaps: 8
       Suppressed. Reusable in:       00:31:40
       Preference will be: 170
```

Meaning Routes with a /16 mask are not impacted by the custom damping rules. Therefore, the default damping rules are in effect.

To repeat, the custom rules are as follows:

- Damp all prefixes with a mask length equal to or greater than 17 more aggressively than routes with a mask length between 9 and 16.
- Damp routes with a mask length between 0 and 8, inclusive, less than routes with a mask length greater than 8.
- Do not damp the 10.128.0.0/9 prefix at all.

Filtering the Damping Information

Purpose Use OR groupings or cascaded piping to simplify the determination of what damping profile is being used for routes with a given mask length.

Action From operational mode, enter the **show route damping suppressed** command.

```
user@R2> show route damping suppressed detail | match "0 announced | damp"

0.0.0.0/0 (1 entry, 0 announced)
               damping-parameters: timid
10.0.0.0/9 (1 entry, 0 announced)
               Default damping parameters used
               damping-parameters: aggressive
               damping-parameters: aggressive
10.224.0.0/11 (1 entry, 0 announced)
               Default damping parameters used
172.16.0.0/16 (1 entry, 0 announced)
               Default damping parameters used
172.16.128.0/17 (1 entry, 0 announced)
               damping-parameters: aggressive
172.16.192.0/20 (1 entry, 0 announced)
               damping-parameters: aggressive
192.168.0.1/32 (1 entry, 0 announced)
```

```
                damping-parameters: aggressive
192.168.0.3/32 (1 entry, 0 announced)
                damping-parameters: aggressive
224.0.0.0/7 (1 entry, 0 announced)
                damping-parameters: timid
```

Meaning When you are satisfied that your EBGP routes are correctly associated with a damping profile, you can issue the **clear bgp damping** operational mode command to restore an active status to your damped routes, which will return your connectivity to normal operation.

Example: Configuring BGP Route Flap Damping Based on the MBGP MVPN Address Family

This example shows how to configure an multiprotocol BGP multicast VPN (also called Next-Generation MVPN) with BGP route flap damping.

Requirements

This example uses Junos OS Release 12.2. BGP route flap damping support for MBGP MVPN, specifically, and on an address family basis, in general, is introduced in Junos OS Release 12.2.

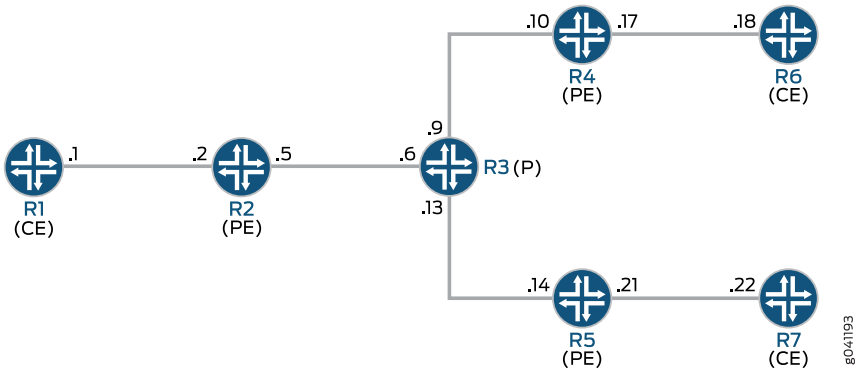
Overview

BGP route flap damping helps to diminish route instability caused by routes being repeatedly withdrawn and readvertised when a link is intermittently failing.

This example uses the default damping parameters and demonstrates an MBGP MVPN scenario with three provider edge (PE) routing devices, three customer edge (CE) routing devices, and one provider (P) routing device.

Figure 38 on page 338 shows the topology used in this example.

Figure 38: MBGP MVPN with BGP Route Flap Damping



On PE Device R4, BGP route flap damping is configured for address family **inet-mvpn**. A routing policy called **dampPolicy** uses the **nlri-route-type** match condition to damp only MVPN route types 3, 4, and 5. All other MVPN route types are not damped.

This example shows the full configuration on all devices in the "CLI Quick Configuration" on page 339 section. The "Configuring Device R4" on page 342 section shows the step-by-step configuration for PE Device R4.

Configuration

CLI Quick Configuration

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

Device R1

```
set interfaces ge-1/2/0 unit 1 family inet address 10.1.1.1/30
set interfaces ge-1/2/0 unit 1 family mpls
set interfaces lo0 unit 1 family inet address 1.1.1.1/32
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.1
set protocols pim rp static address 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.1
```

Device R2

```
set interfaces ge-1/2/0 unit 2 family inet address 10.1.1.2/30
set interfaces ge-1/2/0 unit 2 family mpls
set interfaces ge-1/2/1 unit 5 family inet address 10.1.1.5/30
set interfaces ge-1/2/1 unit 5 family mpls
set interfaces vt-1/2/0 unit 2 family inet
set interfaces lo0 unit 2 family inet address 1.1.1.2/32
set interfaces lo0 unit 102 family inet address 100.1.1.2/32
set protocols mpls interface ge-1/2/1.5
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.2
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 1.1.1.4
set protocols bgp group ibgp neighbor 1.1.1.5
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/1.5
set protocols ldp interface ge-1/2/1.5
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface ge-1/2/0.2
set routing-instances vpn-1 interface vt-1/2/0.2
set routing-instances vpn-1 interface lo0.102
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 provider-tunnel ldp-p2mp
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.102 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/0.2
set routing-instances vpn-1 protocols pim rp static address 100.1.1.2
```

```
set routing-instances vpn-1 protocols pim interface ge-1/2/0.2 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 1.1.1.2
set routing-options autonomous-system 1001
```

Device R3

```
set interfaces ge-1/2/0 unit 6 family inet address 10.1.1.6/30
set interfaces ge-1/2/0 unit 6 family mpls
set interfaces ge-1/2/1 unit 9 family inet address 10.1.1.9/30
set interfaces ge-1/2/1 unit 9 family mpls
set interfaces ge-1/2/2 unit 13 family inet address 10.1.1.13/30
set interfaces ge-1/2/2 unit 13 family mpls
set interfaces lo0 unit 3 family inet address 1.1.1.3/32
set protocols mpls interface ge-1/2/0.6
set protocols mpls interface ge-1/2/1.9
set protocols mpls interface ge-1/2/2.13
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.6
set protocols ospf area 0.0.0.0 interface ge-1/2/1.9
set protocols ospf area 0.0.0.0 interface ge-1/2/2.13
set protocols ldp interface ge-1/2/0.6
set protocols ldp interface ge-1/2/1.9
set protocols ldp interface ge-1/2/2.13
set protocols ldp p2mp
set routing-options router-id 1.1.1.3
```

Device R4

```
set interfaces ge-1/2/0 unit 10 family inet address 10.1.1.10/30
set interfaces ge-1/2/0 unit 10 family mpls
set interfaces ge-1/2/1 unit 17 family inet address 10.1.1.17/30
set interfaces ge-1/2/1 unit 17 family mpls
set interfaces vt-1/2/0 unit 4 family inet
set interfaces lo0 unit 4 family inet address 1.1.1.4/32
set interfaces lo0 unit 104 family inet address 100.1.1.4/32
set protocols rsvp interface all aggregate
set protocols mpls interface all
set protocols mpls interface ge-1/2/0.10
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.4
set protocols bgp group ibgp family inet-vpn unicast
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling damping
set protocols bgp group ibgp neighbor 1.1.1.2 import dampPolicy
set protocols bgp group ibgp neighbor 1.1.1.5
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.4 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.10
set protocols ldp interface ge-1/2/0.10
set protocols ldp p2mp
set policy-options policy-statement dampPolicy term term1 from family inet-mvpn
set policy-options policy-statement dampPolicy term term1 from nlri-route-type 3
set policy-options policy-statement dampPolicy term term1 from nlri-route-type 4
set policy-options policy-statement dampPolicy term term1 from nlri-route-type 5
set policy-options policy-statement dampPolicy term term1 then accept
set policy-options policy-statement dampPolicy then damping no-damp
set policy-options policy-statement dampPolicy then accept
```

```
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set policy-options damping no-damp disable
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/0.4
set routing-instances vpn-1 interface ge-1/2/1.17
set routing-instances vpn-1 interface lo0.104
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.104 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/1.17
set routing-instances vpn-1 protocols pim rp static address 100.1.1.2
set routing-instances vpn-1 protocols pim interface ge-1/2/1.17 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 1.1.1.4
set routing-options autonomous-system 1001
```

Device R5

```
set interfaces ge-1/2/0 unit 14 family inet address 10.1.1.14/30
set interfaces ge-1/2/0 unit 14 family mpls
set interfaces ge-1/2/1 unit 21 family inet address 10.1.1.21/30
set interfaces ge-1/2/1 unit 21 family mpls
set interfaces vt-1/2/0 unit 5 family inet
set interfaces lo0 unit 5 family inet address 1.1.1.5/32
set interfaces lo0 unit 105 family inet address 100.1.1.5/32
set protocols mpls interface ge-1/2/0.14
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.5
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 1.1.1.2
set protocols bgp group ibgp neighbor 1.1.1.4
set protocols ospf area 0.0.0.0 interface lo0.5 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.14
set protocols ldp interface ge-1/2/0.14
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/0.5
set routing-instances vpn-1 interface ge-1/2/1.21
set routing-instances vpn-1 interface lo0.105
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.105 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/1.21
set routing-instances vpn-1 protocols pim rp static address 100.1.1.2
set routing-instances vpn-1 protocols pim interface ge-1/2/1.21 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 1.1.1.5
set routing-options autonomous-system 1001
```

Device R6

```
set interfaces ge-1/2/0 unit 18 family inet address 10.1.1.18/30
set interfaces ge-1/2/0 unit 18 family mpls
```

```
set interfaces lo0 unit 6 family inet address 1.1.1.6/32
set protocols sap listen 224.1.1.1
set protocols ospf area 0.0.0.0 interface lo0.6 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.18
set protocols pim rp static address 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.6
```

Device R7

```
set interfaces ge-1/2/0 unit 22 family inet address 10.1.1.22/30
set interfaces ge-1/2/0 unit 22 family mpls
set interfaces lo0 unit 7 family inet address 1.1.1.7/32
set protocols ospf area 0.0.0.0 interface lo0.7 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.22
set protocols pim rp static address 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.7
```

Configuring Device R4

Step-by-Step Procedure

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

To configure Device R4:

1. Configure the interfaces.

```
[edit interfaces]
user@R4# set ge-1/2/0 unit 10 family inet address 10.1.1.10/30
user@R4# set ge-1/2/0 unit 10 family mpls

user@R4# set ge-1/2/1 unit 17 family inet address 10.1.1.17/30
user@R4# set ge-1/2/1 unit 17 family mpls

user@R4# set vt-1/2/0 unit 4 family inet

user@R4# set lo0 unit 4 family inet address 1.1.1.4/32
user@R4# set lo0 unit 104 family inet address 100.1.1.4/32
```

2. Configure MPLS and the signaling protocols on the interfaces.

```
[edit protocols]
user@R4# set mpls interface all
user@R4# set mpls interface ge-1/2/0.10
user@R4# set rsvp interface all aggregate
user@R4# set ldp interface ge-1/2/0.10
user@R4# set ldp p2mp
```

3. Configure BGP.

   The BGP configuration enables BGP route flap damping for the **inet-mvpn** address family. The BGP configuration also imports into the routing table the routing policy called **dampPolicy**. This policy is applied to neighbor PE Device R2.

```
[edit protocols bgp group ibgp]
user@R4# set type internal
```

```
user@R4# set local-address 1.1.1.4
user@R4# set family inet-vpn unicast
user@R4# set family inet-vpn any
user@R4# set family inet-mvpn signaling damping
user@R4# set neighbor 1.1.1.2 import dampPolicy
user@R4# set neighbor 1.1.1.5
```

4. Configure an interior gateway protocol.

```
[edit protocols ospf]
user@R4# set traffic-engineering

[edit protocols ospf area 0.0.0.0]
user@R4# set interface all
user@R4# set interface lo0.4 passive
user@R4# set interface ge-1/2/0.10
```

5. Configure a damping policy that uses the **nlri-route-type** match condition to damp only MVPN route types 3, 4, and 5.

```
[edit policy-options policy-statement dampPolicy term term1]
user@R4# set from family inet-mvpn
user@R4# set from nlri-route-type 3
user@R4# set from nlri-route-type 4
user@R4# set from nlri-route-type 5
user@R4# set then accept
```

6. Configure the **damping** policy to disable BGP route flap damping.

   The **no-damp** policy (**damping no-damp disable**) causes any damping state that is present in the routing table to be deleted. The **then damping no-damp** statement applies the **no-damp** policy as an action and has no **from** match conditions. Therefore, all routes that are not matched by **term1** are matched by this term, with the result that all other MVPN route types are not damped.

```
[edit policy-options policy-statement dampPolicy]
user@R4# set then damping no-damp
user@R4# set then accept

[edit policy-options]
user@R4# set damping no-damp disable
```

7. Configure the **parent_vpn_routes** to accept all other BGP routes that are not from the **inet-mvpn** address family.

   This policy is applied as an OSPF export policy in the routing instance.

```
[edit policy-options policy-statement parent_vpn_routes]
user@R4# set from protocol bgp
user@R4# set then accept
```

8. Configure the VPN routing and forwarding (VRF) instance.

```
[edit routing-instances vpn-1]
user@R4# set instance-type vrf
user@R4# set interface vt-1/2/0.4
user@R4# set interface ge-1/2/1.17
user@R4# set interface lo0.104
```

```
user@R4# set route-distinguisher 100:100
user@R4# set vrf-target target:1:1
user@R4# set protocols ospf export parent_vpn_routes
user@R4# set protocols ospf area 0.0.0.0 interface lo0.104 passive
user@R4# set protocols ospf area 0.0.0.0 interface ge-1/2/1.17
user@R4# set protocols pim rp static address 100.1.1.2
user@R4# set protocols pim interface ge-1/2/1.17 mode sparse
user@R4# set protocols mvpn
```

9. Configure the router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@R4# set router-id 1.1.1.4
user@R4# set autonomous-system 1001
```

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

```
user@R4# commit
```

Results

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

```
user@R4# show interfaces
ge-1/2/0 {
    unit 10 {
        family inet {
            address 10.1.1.10/30;
        }
        family mpls;
    }
}
ge-1/2/1 {
    unit 17 {
        family inet {
            address 10.1.1.17/30;
        }
        family mpls;
    }
}
vt-1/2/0 {
    unit 4 {
        family inet;
    }
}
lo0 {
    unit 4 {
        family inet {
            address 1.1.1.4/32;
        }
    }
    unit 104 {
        family inet {
```

```
        address 100.1.1.4/32;
      }
    }
}
user@R4# show protocols
rsvp {
  interface all {
    aggregate;
  }
}
mpls {
  interface all;
  interface ge-1/2/0.10;
}
bgp {
  group ibgp {
    type internal;
    local-address 1.1.1.4;
    family inet-vpn {
      unicast;
      any;
    }
    family inet-mvpn {
      signaling {
        damping;
      }
    }
    neighbor 1.1.1.2 {
      import dampPolicy;
    }
    neighbor 1.1.1.5;
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface lo0.4 {
      passive;
    }
    interface ge-1/2/0.10;
  }
}
ldp {
  interface ge-1/2/0.10;
  p2mp;
}
user@R4# show policy-options
policy-statement dampPolicy {
  term term1 {
    from {
      family inet-mvpn;
      nlri-route-type [ 3 4 5 ];
    }
    then accept;
```

```
    }
    then {
      damping no-damp;
      accept;
    }
  }
  policy-statement parent_vpn_routes {
    from protocol bgp;
    then accept;
  }
  damping no-damp {
    disable;
  }
```

```
user@R4# show routing-instances
vpn-1 {
    instance-type vrf;
    interface vt-1/2/0.4;
    interface ge-1/2/1.17;
    interface lo0.104;
    route-distinguisher 100:100;
    vrf-target target:1:1;
    protocols {
        ospf {
            export parent_vpn_routes;
            area 0.0.0.0 {
                interface lo0.104 {
                    passive;
                }
                interface ge-1/2/1.17;
            }
        }
        pim {
            rp {
                static {
                    address 100.1.1.2;
                }
            }
            interface ge-1/2/1.17 {
                mode sparse;
            }
        }
        mvpn;
    }
}
```

```
user@R4# show routing-optons
router-id 1.1.1.4;
autonomous-system 1001;
```

Verification

Confirm that the configuration is working properly.

- Verifying That Route Flap Damping Is Disabled on page 347
- Verifying Route Flap Damping on page 347

Verifying That Route Flap Damping Is Disabled

Purpose Verify the presence of the **no-damp** policy, which disables damping for MVPN route types other than 3, 4, and 5.

Action From operational mode, enter the **show policy damping** command.

```
user@R4> show policy damping
Default damping information:
  Halflife: 15 minutes
  Reuse merit: 750 Suppress/cutoff merit: 3000
  Maximum suppress time: 60 minutes
  Computed values:
    Merit ceiling: 12110
    Maximum decay: 6193
Damping information for "no-damp":
 Damping disabled
```

Meaning The output shows that the default damping parameters are in effect and that the **no-damp** policy is also in effect for the specified route types.

Verifying Route Flap Damping

Purpose Check whether BGP routes have been damped.

Action From operational mode, enter the **show bgp summary** command.

```
user@R4> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0
                       6          6          0          0          0          0
bgp.l3vpn.2
                       0          0          0          0          0          0
bgp.mvpn.0
                       2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
1.1.1.2                1001       3159       3155       0       0    23:43:47
Establ
  bgp.l3vpn.0: 3/3/3/0
  bgp.l3vpn.2: 0/0/0/0
  bgp.mvpn.0: 1/1/1/0
  vpn-1.inet.0: 3/3/3/0
  vpn-1.mvpn.0: 1/1/1/0
1.1.1.5                1001       3157       3154       0       0    23:43:40
Establ
  bgp.l3vpn.0: 3/3/3/0
  bgp.l3vpn.2: 0/0/0/0
  bgp.mvpn.0: 1/1/1/0
  vpn-1.inet.0: 3/3/3/0
  vpn-1.mvpn.0: 1/1/1/0
```

Meaning The Damp State field shows that zero routes in the bgp.mvpn.0 routing table have been damped. Further down, the last number in the State field shows that zero routes have been damped for BGP peer 1.1.1.2.

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

CHAPTER 10

BGP Monitoring Configuration

Example: Configuring BGP Trace Operations

Understanding Trace Operations for BGP Protocol Traffic

You can trace various BGP protocol traffic to help you debug BGP protocol issues. To trace BGP protocol traffic, include the **traceoptions** statement at the **[edit protocols bgp]** hierarchy level. For routing instances, include the **traceoptions** statement at the **[edit routing-instances *routing-instance-name* protocols bgp]** hierarchy level.

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can specify the following BGP protocol-specific trace options using the **flag** statement:

- **4byte-as**—4-byte AS events.
- **bfd**—BFD protocol events.
- **damping**—Damping operations.
- **graceful-restart**—Graceful restart events.
- **keepalive**—BGP keepalive messages.
- **nsr-synchronization**—Nonstop active routing synchronization events.
- **open**—BGP open packets. These packets are sent between peers when they are establishing a connection.
- **packets**—All BGP protocol packets.
- **refresh**—BGP refresh packets.
- **update**—BGP update packets. These packets provide routing updates to BGP systems.

Global tracing options are inherited from the configuration set by the **traceoptions** statement at the **[edit routing-options]** hierarchy level. You can override the following global trace options for the BGP protocol using the **traceoptions flag** statement included at the **[edit protocols bgp]** hierarchy level:

- **all**—All tracing operations
- **general**—All normal operations and routing table changes (a combination of the normal and route trace operations)
- **normal**—Normal events
- **policy**—Policy processing
- **route**—Routing information
- **state**—State transitions
- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing

You can optionally specify one or more of the following flag modifiers:

- **detail**—Detailed trace information.
- **filter**—Filter trace information. Applies only to **route** and **damping** tracing flags.
- **receive**—Packets being received.
- **send**—Packets being transmitted.

NOTE: Use the **all** trace flag and the **detail** flag modifier with caution because these might cause the CPU to become very busy.

NOTE: If you only enable the **update** flag, received keepalive messages do not generate a trace message.

You can filter trace statements and display only the statement information that passes through the filter by specifying the **filter** flag modifier. The **filter** modifier is only supported for the **route** and **damping** tracing flags.

The **match-on** statement specifies filter matches based on prefixes. It is used to match on route filters.

NOTE: Per-neighbor trace filtering is not supported on a BGP per-neighbor level for **route** and **damping** flags. Trace option filtering support is on a peer group level.

Example: Viewing BGP Trace Files on Logical Systems

This example shows how to list and view files that are stored on a logical system.

Requirements

- You must have the **view** privilege for the logical system.
- Configure a network, such as the BGP network shown in "Example: Configuring Internal BGP Peering Sessions on Logical Systems" on page 46.

Overview

Logical systems have their individual directory structure created in the **/var/logical-systems/*logical-system-name*** directory. It contains the following subdirectories:

- **/config**—Contains the active configuration specific to the logical system.
- **/log**—Contains system log and tracing files specific to the logical system.

  To maintain backward compatibility for the log files with previous versions of Junos OS, a symbolic link (symlink) from the **/var/logs/*logical-system-name*** directory to the **/var/logical-systems/*logical-system-name*** directory is created when a logical system is configured.
- **/tmp**—Contains temporary files specific to the logical system.

The file system for each logical system enables logical system users to view trace logs and modify logical system files. Logical system administrators have full access to view and modify all files specific to the logical system.

Logical system users and administrators can save and load configuration files at the logical-system level using the **save** and **load** configuration mode commands. In addition, they can also issue the **show log**, **monitor**, and **file** operational mode commands at the logical-system level.

This example shows how to configure and view a BGP trace file on a logical system. 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 logical-systems A protocols bgp group internal-peers traceoptions file bgp-log
set logical-systems A protocols bgp group internal-peers traceoptions file size 10k
set logical-systems A protocols bgp group internal-peers traceoptions file files 2
set logical-systems A protocols bgp group internal-peers traceoptions flag update detail
```

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 on the logical system.

   ```
   [edit logical-systems A protocols bgp group internal-peers]
   user@host# set traceoptions file bgp-log
   user@host# set traceoptions file size 10k
   user@host# set traceoptions file files 2
   user@host# set traceoptions flag update detail
   ```

2. 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 on the main router, list the directories on the logical system.

   ```
   user@host> file list /var/logical-systems/A
   /var/logical-systems/A:
   config/
   log/
   tmp/
   ```

2. In operational mode on the main router, list the log files on the logical system.

   ```
   user@host> file list /var/logical-systems/A/log/
   /var/logical-systems/A/log:
   bgp-log
   ```

3. View the contents of the **bgp-log** file.

```
user@host> file show /var/logical-systems/A/log/bgp-log
Aug 10 17:12:01 trace_on: Tracing to "/var/log/A/bgp-log" started
Aug 10 17:14:22.826182 bgp_peer_mgmt_clear:5829: NOTIFICATION sent to
192.163.6.4 (Internal AS 17): code 6 (Cease) subcode 4 (Administratively
Reset), Reason: Management session cleared BGP neighbor
Aug 10 17:14:22.826445 bgp_send: sending 21 bytes to 192.163.6.4 (Internal
AS 17)
Aug 10 17:14:22.826499
Aug 10 17:14:22.826499 BGP SEND 192.168.6.5+64965 -> 192.163.6.4+179
Aug 10 17:14:22.826559 BGP SEND message type 3 (Notification) length 21
Aug 10 17:14:22.826598 BGP SEND Notification code 6 (Cease) subcode 4
(Administratively Reset)
Aug 10 17:14:22.831756 bgp_peer_mgmt_clear:5829: NOTIFICATION sent to
192.168.40.4 (Internal AS 17): code 6 (Cease) subcode 4 (Administratively
Reset), Reason: Management session cleared BGP neighbor
Aug 10 17:14:22.831851 bgp_send: sending 21 bytes to 192.168.40.4 (Internal
 AS 17)
Aug 10 17:14:22.831901
Aug 10 17:14:22.831901 BGP SEND 192.168.6.5+53889 -> 192.168.40.4+179
Aug 10 17:14:22.831959 BGP SEND message type 3 (Notification) length 21
Aug 10 17:14:22.831999 BGP SEND Notification code 6 (Cease) subcode 4
(Administratively Reset)
...
```

4. Filter the output of the log file.

```
user@host> file show /var/logical-systems/A/log/bgp-log | match "flags 0x40"
Aug 10 17:14:54.867460 BGP SEND flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.867595 BGP SEND flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.867650 BGP SEND flags 0x40 code NextHop(3): 192.168.6.5
Aug 10 17:14:54.867692 BGP SEND flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.884529 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.884581 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.884628 BGP RECV flags 0x40 code NextHop(3): 192.163.6.4
Aug 10 17:14:54.884667 BGP RECV flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.911377 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.911422 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.911466 BGP RECV flags 0x40 code NextHop(3): 192.168.40.4
Aug 10 17:14:54.911507 BGP RECV flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.916008 BGP SEND flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.916054 BGP SEND flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.916100 BGP SEND flags 0x40 code NextHop(3): 192.168.6.5
Aug 10 17:14:54.916143 BGP SEND flags 0x40 code LocalPref(5): 100
Aug 10 17:14:54.920304 BGP RECV flags 0x40 code Origin(1): IGP
Aug 10 17:14:54.920348 BGP RECV flags 0x40 code ASPath(2) length 0: <null>
Aug 10 17:14:54.920393 BGP RECV flags 0x40 code NextHop(3): 10.0.0.10
Aug 10 17:14:54.920434 BGP RECV flags 0x40 code LocalPref(5): 100
```

5. View the tracing operations in real time.

```
user@host> clear bgp neighbor logical-system A
Cleared 2 connections
```

CAUTION: Clearing the BGP neighbor table is disruptive in a production environment.

6. Run the **monitor start** command with an optional **match** condition.

```
user@host> monitor start A/bgp-log | match 0.0.0.0/0
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
```

7. Pause the **monitor** command by pressing Esc-Q.
   To unpause the output, press Esc-Q again.

8. Halt the **monitor** command by pressing Enter and typing **monitor stop**.

```
[Enter]
user@host> monitor stop
```

9. When you are finished troubleshooting, consider deactivating trace logging to avoid any unnecessary impact to system resources.

```
[edit protocols bgp group internal-peers]
user@host:A# deactivate traceoptions
user@host:A# commit
```

   When configuration is deactivated, it appears in the configuration with the **inactive** tag.To reactivate trace operations, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# show

type internal;
inactive: traceoptions {
    file bgp-log size 10k files 2;
    flag update detail;
    flag all;
}
local-address 192.168.6.5;
export send-direct;
neighbor 192.163.6.4;
neighbor 192.168.40.4;
```

10. To reactivate trace operations, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# activate traceoptions
user@host:A# commit
```

Deactivating and Reactivating Trace Logging

Step-by-Step Procedure

To deactivate and reactivate the trace file:

1. When you are finished troubleshooting, consider deactivating trace logging to avoid an unnecessary impact to system resources.

```
[edit protocols bgp group internal-peers]
user@host:A# deactivate traceoptions
user@host:A# commit
```

   When configuration is deactivated, the statement appears in the configuration with the **inactive** tag.

```
[edit protocols bgp group internal-peers]
user@host:A# show

type internal;
inactive: traceoptions {
    file bgp-log size 10k files 2;
    flag update detail;
    flag all;
}
local-address 192.168.6.5;
export send-direct;
neighbor 192.163.6.4;
neighbor 192.168.40.4;
```

2. To reactivate logging, use the **activate** configuration-mode statement.

```
[edit protocols bgp group internal-peers]
user@host:A# activate traceoptions
user@host:A# commit
```

Results

From configuration mode, confirm your configuration by entering the **show logical-systems A protocols bgp group internal-peers** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show logical-systems A protocols bgp group internal-peers
traceoptions {
  file bgp-log size 10k files 2;
  flag update detail;
}
```

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:A> show log bgp-log
Aug 12 11:20:57 trace_on: Tracing to "/var/log/A/bgp-log" started
```

Related Documentation

- Understanding External BGP Peering Sessions on page 11
- *BGP Configuration Overview*

CHAPTER 11

Configuration Statements

accept-remote-nexthop

Syntax accept-remote-nexthop;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Specify that a single-hop EBGP peer accepts a remote next hop with which it does not share a common subnet. Configure a separate import policy on the EBGP peer to specify the remote next hop. You cannot configure **multihop** and **accept-remote-nexthop** statements for the same EBGP peer.

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

Related Documentation
- Example: Configuring Single-Hop EBGP Peers to Accept Remote Next Hops on page 237
- Understanding Route Advertisement on page 173
- *multipath*

advertise-external

Syntax advertise-external {*conditional*};

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *neighbor-address*]

Release Information Statement introduced in Junos OS Release 9.3.
Statement introduced in Junos OS Release 9.3 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Specify BGP to advertise the best external route into an IBGP mesh group, a route reflector cluster, or an AS confederation even if the best route is an internal route.

In general, deployed BGP implementations do not advertise the external route with the highest local preference value to internal peers unless it is the best route. Although this behavior was required by an earlier version of the BGP version 4 specification, RFC 1771, it was typically not followed in order to minimize the amount of advertised information and to prevent routing loops. However, there are scenarios in which advertising the best external route is beneficial, in particular, situations that can result in IBGP route oscillation.

The **advertise-external** statement is supported at both the group and neighbor level. If you configure the statement at the neighbor level, you must configure it for all neighbors in a group. Otherwise, the group is automatically split into different groups.

In a confederation, when advertising a route to a confederation border router, any route from a different confederation sub-AS is considered external. When configuring the **advertise-external** statement for an AS confederation, it is recommended that EBGP peers belonging to different autonomous systems are configured in a separate EBGP peer group. This ensures consistency while BGP sends the best external route to peers in the configured peer group.

To configure the **advertise-external** statement on a route reflector, you must disable intracluster reflection with the **no-client-reflect** statement.

When a routing device is configured as a route reflector for a cluster, a route advertised by the route reflector is considered internal if it is received from an internal peer with the same cluster identifier or if both peers have no cluster identifier configured. A route received from an internal peer that belongs to another cluster, that is, with a different cluster identifier, is considered external.

The **conditional** option causes BGP to advertise the external route only if the route selection process reaches the point where the multiple exit discriminator (MED) metric

is evaluated. As a result, an external route with an AS path longer than that of the active path is not advertised.

Junos OS also provides support for configuring a BGP export policy that matches on the state of an advertised route. You can match on either active or inactive routes.

Default BGP does not advertise the external route with the highest local preference value to internal peers unless it is the best route.

Options

conditional—(Optional) Advertise the best external path only if the route selection process reaches the point at which the multiple exit discriminator (MED) metric is evaluated. The **conditional** option restricts advertisement to when the best external path and the active path are equal until the MED step of the route selection process. This implies that external routes with a longer AS path length than the active path, for instance, are not advertised. The criteria used for selecting the best external path is the same whether or not the **conditional** option is configured.

Required Privilege Level

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

Related Documentation

- *Example: Configuring BGP to Advertise the Best External Route to Internal Peers*
- advertise-inactive on page 362

advertise-inactive

Syntax advertise-inactive;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure the routing table to export to BGP the best route learned by BGP even if Junos OS did not select this route to be an active route.

One way to achieve multivendor compatibility is to include the **advertise-inactive** statement in the external BGP (EBGP) configuration. By default, BGP stores the route information it receives from update messages in the Junos OS routing table, and the routing table exports only active routes into BGP, which BGP then advertises to its peers. The **advertise-inactive** statement causes Junos OS to advertise the best BGP route that is inactive because of IGP preference. When you use the **advertise-inactive** statement, the Junos OS device uses, for example, the OSPF route for forwarding, and the other vendor's device uses the EBGP route for forwarding. However, from the perspective of an EBGP peer in a neighboring AS, both vendors' devices appear to behave the same way.

Default By default, BGP stores the route information it receives from update messages in the Junos OS routing table, and the routing table exports only active routes into BGP, which BGP then advertises to its peers.

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

Related Documentation
- *Example: Setting BGP to Advertise Inactive Routes*
- Example: Configuring the Preference Value for BGP Routes on page 191
- Example: Configuring BGP Route Preference (Administrative Distance) on page 190

- advertise-external on page 360

advertise-peer-as

Syntax advertise-peer-as;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Disable the default behavior of suppressing AS routes.

If you include the **advertise-peer-as** statement in the configuration, BGP advertises routes learned from one external BGP (EBGP) peer back to another EBGP peer in the same autonomous system (AS).

Another way to disable the route suppression default behavior is with the **as-override** statement. If you include both the **as-override** and **no-advertise-peer-as** statements in the configuration, the **no-advertise-peer-as** statement is ignored.

Default By default, Junos OS does not advertise the routes learned from one EBGP peer back to the same external BGP (EBGP) peer. In addition, the software does not advertise those routes back to any EBGP peers that are in the same AS as the originating peer, regardless of the routing instance.

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

Related Documentation
- *Example: Disabling Suppression of Route Advertisements*
- *Example: Configuring a Layer 3 VPN with Route Reflection and AS Override*
- no-advertise-peer-as on page 435

algorithm (BGP BFD Authentication)

Syntax algorithm *algorithm-name*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication],
[edit protocols bgp bgp bfd-liveness-detection authentication],
[edit protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols bgp bgp bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication]

Release Information
Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure the algorithm used to authenticate the specified BFD session.

Options *algorithm-name*—Authentication algorithm name: **simple-password**, **keyed-md5**, **keyed-sha-1**, **meticulous-keyed-md5**, **meticulous-keyed-sha-1**.

simple-password—Plain-text password. One to 16 bytes of plain text are used to authenticate the BFD session. One or more passwords can be configured. This method is the least secure and should be used only when BFD sessions are not subject to packet interception.

keyed-md5—Keyed Message Digest 5 hash algorithm for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed MD5 uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than or equal to the last sequence number received. Although more secure than a simple password, this method is vulnerable to replay attacks. Increasing the rate at which the sequence number is updated can reduce this risk.

meticulous-keyed-md5—Meticulous keyed Message Digest 5 hash algorithm. This method works in the same manner as keyed MD5, but the sequence number is updated with every packet. Although more secure than keyed MD5 and simple passwords, this method can take additional time to authenticate the session.

keyed-sha-1—Keyed Secure Hash Algorithm I for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed SHA uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. The key is not carried within the packets. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than the last sequence number received.

meticulous-keyed-sha-1—Meticulous keyed Secure Hash Algorithm I. This method works in the same manner as keyed SHA, but the sequence number is updated with every packet. Although more secure than keyed SHA and simple passwords, this method can take additional time to authenticate the session.

Required Privilege Level

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

Related Documentation

- *Example: Configuring BFD Authentication for Static Routes*
- Example: Configuring BGP Route Authentication on page 299
- Example: Configuring EBGP Multihop Sessions on page 181
- authentication on page 367
- bfd-liveness-detection on page 372
- key-chain on page 405
- loose-check on page 416

apply-groups

Syntax apply-groups [*group-names*];

Hierarchy Level All hierarchy levels

Release Information Statement introduced before Junos OS Release 7.4.

Description Apply a configuration group to a specific hierarchy level in a configuration, to have a configuration inherit the statements in the configuration group.

You can specify more than one group name. You must list them in order of inheritance priority. The configuration data in the first group takes priority over the data in subsequent groups.

Options ***group-names***—One or more names specified in the **groups** statement.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation
- *Applying a Junos Configuration Group*
- *groups*

apply-groups-except

Syntax apply-groups-except [*group-names*];

Hierarchy Level All hierarchy levels except the top level

Release Information Statement introduced before Junos OS Release 7.4.

Description Disable inheritance of a configuration group.

Options ***group-names***—One or more names specified in the **groups** statement.

Required Privilege Level configure—To enter configuration mode, but other required privilege levels depend on where the statement is located in the configuration hierarchy.

Related Documentation
- *groups*
- *Disabling Inheritance of a Junos OS Configuration Group*

authentication (BGP BFD Liveness Detection)

Syntax

```
authentication {
  algorithm algorithm-name;
  key-chain key-chain-name;
  loose-check ;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit protocols bgp bgp bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]

Release Information

Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Specify the router and route authentication to mitigate the risk of being attacked by a machine or router that has been configured to share incorrect routing information with another router. Router and route authentication enables routers to share information only if they can verify that they are talking to a trusted source, based on a password (key). In this method, a hashed key is sent along with the route being sent to another router. The receiving router compares the sent key to its own configured key. If they are the same, the receiving router accepts the route.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- *Example: Configuring BFD for Static Routes*
- *Example: Configuring BFD Authentication for Static Routes*
- Example: Configuring BGP Route Authentication on page 299
- algorithm on page 364

- bfd-liveness-detection on page 372
- key-chain on page 405
- loose-check on page 416

authentication-algorithm

Syntax authentication-algorithm *algorithm*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* protocols ldp session *session-address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ldp session *session-address*],
[edit logical-systems *logical-system-name* routing-options bmp],
[edit logical-systems *logical-system-name* routing-options bmp station *station-name*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit protocols ldp session *session-address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols ldp session *session-address*],
[edit routing-options bmp],
[edit routing-options bmp station *station-name*]

Release Information
Statement introduced in Junos OS Release 7.6.
Statement introduced for BGP in Junos OS Release 8.0.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced for BMP in Junos OS Release 13.2X51-D15 for the QFX Series.
Statement introduced for BMP in Junos OS Release 13.3.

Description Configure an authentication algorithm type.

Options ***algorithm***—Specify one of the following types of authentication algorithms:

- **aes-128-cmac-96**—Cipher-based message authentication code (AES128, 96 bits).
- **hmac-sha-1-96**—Hash-based message authentication code (SHA1, 96 bits).
- **md5**—Message digest 5.

Default: hmac-sha-1-96

NOTE: The default is not displayed in the output of the **show bgp bmp** command unless a key or key-chain is also configured.

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

Related Documentation
- Example: Configuring Route Authentication for BGP on page 300
- *Configuring BGP Monitoring Protocol Version 3*

authentication-key (Protocols BGP and BMP)

Syntax
authentication-key *key*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-options bmp],
[edit logical-systems *logical-system-name* routing-options bmp station *station-name*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit routing-options bmp],
[edit routing-options bmp station *station-name*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced for BMP in Junos OS Release 13.2X51-D15 for the QFX Series.
Statement introduced for BMP version 3 in Junos OS Release 13.3.

Description
Configure an MD5 authentication key (password). Neighboring routing devices use the same password to verify the authenticity of BGP packets sent from this system.

Options
key—Authentication password. It can be up to 126 characters. Characters can include any ASCII strings. If you include spaces, enclose all characters in quotation marks (“ ”).

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

Related Documentation
- Example: Configuring Route Authentication for BGP on page 300
- *Configuring BGP Monitoring Protocol Version 3*

authentication-key-chain (Protocols BGP and BMP)

Syntax authentication-key-chain *key-chain*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-options bmp],
[edit logical-systems *logical-system-name* routing-options bmp station *station-name*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit routing-options bmp],
[edit routing-options bmp station *station-name*]

Release Information
Statement introduced in Junos OS Release 8.0.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced for BMP in Junos OS Release 13.2X51-D15 for the QFX Series.
Statement introduced for BMP in Junos OS Release 13.3.

Description Apply and enable an authentication keychain to the routing device. Note that the referenced key chain must be defined. When configuring the authentication key update feature for BGP, you cannot commit the **0.0.0.0/allow** statement with authentication keys or key chains. The CLI issues a warning and fails to commit the configuration.

Options
key-chain—Authentication keychain name. It can be up to 126 characters. Characters can include any ASCII strings. If you include spaces, enclose all characters in quotation marks (“ ”).

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

Related Documentation
- Example: Configuring Route Authentication for BGP on page 300
- *Example: Configuring BFD Authentication for Static Routes*
- *Configuring the Authentication Key Update Mechanism for BGP and LDP Routing Protocols*
- *Configuring BGP Monitoring Protocol Version 3*

bfd-liveness-detection (Protocols BGP)

Syntax

```
bfd-liveness-detection {
  authentication {
    algorithm algorithm-name;
    key-chain key-chain-name;
    loose-check;
  }
  detection-time {
    threshold milliseconds;
  }
  hold-down-interval milliseconds;
  minimum-interval milliseconds;
  minimum-receive-interval milliseconds;
  multiplier number;
  no-adaptation;
  session-mode (automatic | multihop | single-hop);
  transmit-interval {
    minimum-interval milliseconds;
    threshold milliseconds;
  }
  version (1 | automatic);
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information

Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
detection-time threshold and **transmit-interval threshold** options introduced in Junos OS Release 8.2
Support for logical routers introduced in Junos OS Release 8.3.
Support for IBGP and multihop EBGP sessions introduced in Junos OS Release 8.3.
holddown-interval statement introduced in Junos OS Release 8.5. You can configure this statement only for EBGP peers at the **[edit protocols bgp group *group-name* neighbor *address*]** hierarchy level.
no-adaptation statement introduced in Junos OS Release 9.0.
Support for BFD authentication introduced in Junos OS Release 9.6.

Support for BFD on IPv6 interfaces with BGP introduced in Junos OS Release 11.2. Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure bidirectional failure detection (BFD) timers and authentication for BGP.

For IBGP and multihop EBGP support, configure the **bfd-liveness-detection** statement at the global **[edit bgp protocols]** hierarchy level. You can also configure IBGP and multihop support for a routing instance or a logical system.

Options

authentication algorithm ***algorithm-name*** (Optional)—Configure the algorithm used to authenticate the specified BFD session: **simple-password**, **keyed-md5**, **keyed-sha-1**, **meticulous-keyed-md5**, **meticulous-keyed-sha-1**.

authentication key-chain ***key-chain-name*** (Optional)—Associate a security key with the specified BFD session using the name of the security keychain. The keychain name must match one of the keychains configured in the **authentication-key-chains key-chain** statement at the **[edit security]** hierarchy level.

authentication loose-check—(Optional) Configure loose authentication checking on the BFD session. Use only for transitional periods when authentication may not be configured at both ends of the BFD session.

detection-time threshold ***milliseconds*** (Optional)—Configure a threshold. When the BFD session detection time adapts to a value equal to or greater than the threshold, a single trap and a single system log message are sent.

holddown-interval ***milliseconds*** (Optional)—Configure an interval specifying how long a BFD session must remain up before a state change notification is sent. When you configure the hold-down interval for the BFD protocol for EBGP, the BFD session is unaware of the BGP session during this time. In this case, if the BGP session goes down during the configured hold-down interval, BFD already assumes it is down and does not send a state change notification. The **holddown-interval** statement is supported only for EBGP peers at the **[edit protocols bgp group** ***group-name*** **neighbor** ***address*****]** hierarchy level. If the BFD session goes down and then comes back up during the configured hold-down interval, the timer is restarted. You must configure the hold-down interval on both EBGP peers. If you configure the hold-down interval for a multihop EBGP session, you must also configure a local IP address by including the **local-address** statement at the **[edit protocols bgp group** ***group-name*****]** hierarchy level.

Range: 0 through 255,000

Default: 0

minimum-interval ***milliseconds*** (Required)—Configure the minimum intervals at which the local routing device transmits hello packets and then expects to receive a reply from a neighbor with which it has established a BFD session. This value represents the minimum interval at which the local routing device transmits hello packets as well as the minimum interval that the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds. Optionally, instead of using this statement, you can specify the minimum transmit and receive intervals separately (using the **minimum-receive-interval** and **transmit-interval minimal-interval** statements).

Range: 1 through 255,000

minimum-receive-interval ***milliseconds*** (Optional)— Configure only the minimum interval at which the local routing device expects to receive a reply from a neighbor with which it has established a BFD session.

Range: 1 through 255,000

multiplier ***number*** (Optional)—Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down.

Range: 1 through 255

Default: 3

no-adaptation (Optional)—Configure BFD sessions not to adapt to changing network conditions. We recommend that you not disable BFD adaptation unless it is preferable to not to have BFD adaptation enabled in your network.

transmit-interval threshold ***milliseconds*** (Optional)—Configure a threshold. When the BFD session transmit interval adapts to a value greater than the threshold, a single trap and a single system message are sent. The interval threshold must be greater than the minimum transmit interval.

Range: 0 through 4,294,967,295 ($2^{32} - 1$)

transmit-interval minimum-interval ***milliseconds*** (Optional)—Configure only the minimum interval at which the local routing device transmits hello packets to a neighbor with which it has established a BFD session.

Range: 1 through 255,000

version (Optional)—Configure the BFD version to detect.

Range: **1** or **automatic** (autodetect the BFD version)

Default: **automatic**

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- *Example: Configuring BFD for Static Routes*
- *Example: Configuring BFD Authentication for Static Routes*
- Example: Configuring BFD on Internal BGP Peer Sessions on page 216
- Example: Configuring BFD Authentication for BGP on page 226
- Understanding BFD for BGP on page 215

bgp

Syntax bgp { ... }

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Enable BGP on the routing device or for a routing instance.

Default BGP is disabled.

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

Related Documentation

- *BGP Feature Guide for Routing Devices*

bgp-orf-cisco-mode

Syntax bgp-orf-cisco-mode;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp outbound-route-filter],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* outbound-route-filter],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* outbound-route-filter],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp outbound-route-filter],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* outbound-route-filter,
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* outbound-route-filter],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options outbound-route-filter],
[edit logical-systems *logical-system-name* routing-options outbound-route-filter],
[edit protocols bgp outbound-route-filter],
[edit protocols bgp group *group-name* outbound-route-filter],
[edit protocols bgp group *group-name* neighbor *address* outbound-route-filter],
[edit routing-instances *routing-instance-name* protocols bgp outbound-route-filter],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* outbound-route-filter],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* outbound-route-filter],
[edit routing-instances *routing-instance-name* routing-options outbound-route-filter],
[edit routing-options outbound-route-filter]

Release Information
Statement introduced in Junos OS Release 9.2.
Statement introduced in Junos OS Release 9.2 for EX Series switches.
Support for the BGP group and neighbor hierarchy levels introduced in Junos OS Release 9.2.
Support for the BGP group and neighbor hierarchy levels introduced in Junos OS Release 9.3 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

Description Enable interoperability with routing devices that use the vendor-specific outbound route filter compatibility code of 130 and code type of 128.

NOTE: To enable interoperability for all BGP peers configured on the routing device, include the statement at the **[edit routing-options outbound-route-filter]** hierarchy level.

Default Disabled

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

Related Documentation

- Example: Configuring BGP Prefix-Based Outbound Route Filtering on page 177

cluster

Syntax cluster *cluster-identifier*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Specify the cluster identifier to be used by the route reflector cluster in an internal BGP group.

CAUTION:

If you configure both route reflection and VPNs on the same routing device, the following modifications to the route reflection configuration cause current BGP sessions to be reset:

- Adding a cluster ID—If a BGP session shares the same AS number with the group where you add the cluster ID, all BGP sessions are reset regardless of whether the BGP sessions are contained in the same group.
- Creating a new route reflector—If you have an IBGP group with an AS number and create a new route reflector group with the same AS number, all BGP sessions in the IBGP group and the new route reflector group are reset.

NOTE: If you change the address family specified in the **[edit protocols bgp family]** hierarchy level, all current BGP sessions on the routing device are dropped and then reestablished.

Options ***cluster-identifier***—4-byte identifier (such as an IPv4 address).

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

Related Documentation

- Example: Configuring BGP Route Reflectors on page 275
- Understanding External BGP Peering Sessions on page 11
- no-client-reflect on page 437

damping (Protocols BGP)

Syntax damping;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp family *family*],
[edit logical-systems *logical-system-name* protocols bgp family *family*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family *family*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* family *family*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* family *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp family *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp family *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* family *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* family *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* family *family*],
[edit protocols bgp],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* family *family*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp group *group-name* neighbor *address* family *family*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp family *family*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* family *family*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* family *family*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Support for flap damping at the address family level introduced in Junos OS Release 12.2.

Description Enable route flap damping. BGP route flapping describes the situation in which BGP systems send an excessive number of update messages to advertise network reachability information. Flap damping reduces the number of update messages sent between BGP

peers, thereby reducing the load on these peers, without adversely affecting the route convergence time for stable routes.

You typically apply flap damping to external BGP (EBGP) routes (that is, to routes in different ASs). You can also apply it within a confederation, between confederation member ASs. Because routing consistency within an AS is important, do not apply flap damping to internal BGP (IBGP) routes. (If you do, it is ignored.) The exception to this rule is when flap damping is applied at the address family level. When you apply flap damping at the address family level, it works for both IBGP and EBGP.

Default Flap damping is disabled on the routing device.

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

Related Documentation

- Examples: Configuring BGP Flap Damping on page 328
- Example: Configuring BGP Route Flap Damping Based on the MBGP MVPN Address Family on page 338

description (Protocols BGP)

Syntax description *text-description*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Provide a description of the global, group, or neighbor configuration. If the text includes one or more spaces, enclose it in quotation marks (" "). The test is displayed in the output of the **show** command and has no effect on the configuration.

Options ***text-description***—Text description of the configuration. It is limited to 255 characters.

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

Related Documentation
- *BGP Feature Guide for Routing Devices*

detection-time (BFD Liveness Detection)

Syntax

```
detection-time {
    threshold milliseconds;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection],
[edit protocols bgp bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection]

Release Information

Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPNs and VPLS.

Description

Enable BFD failure detection. The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance

is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the **clear bfd adaptation** command to return BFD interval timers to their configured values. The **clear bfd adaptation** command is hitless, meaning that the command does not affect traffic flow on the routing device.

The remaining statement is explained separately.

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

Related Documentation
- *Configuring BFD for Layer 2 VPN and VPLS*
- Example: Configuring BFD for BGP on page 215
- bfd-liveness-detection on page 372
- threshold on page 453

disable (Protocols BGP)

Syntax
disable;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description
Disable BGP on the system.

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

disable (BGP Graceful Restart)

Syntax disable;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp graceful-restart],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* graceful-restart],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* graceful-restart],
[edit protocols bgp graceful-restart],
[edit protocols bgp group *group-name* graceful-restart],
[edit protocols bgp group *group-name* neighbor *address* graceful-restart]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Disable graceful restart for BGP. Graceful restart allows a routing device undergoing a restart to inform its adjacent neighbors and peers of its condition.

NOTE: When you disable graceful restart at one level in the configuration statement hierarchy, it is also disabled at lower levels in the same hierarchy. For example, if you disable graceful restart at the **[edit protocols bgp group *group-name*]** hierarchy level, it is disabled for all the peers in the group. Therefore, if you want to enable graceful restart for some peers in a group and disable it for others, enable graceful restart at the **[edit protocols bgp group *group-name*]** hierarchy level and disable graceful restart for each peer at the **[edit protocols bgp group *group-name* neighbor *address*]** hierarchy level.

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

Related Documentation
- *Configuring Graceful Restart Options for BGP*
- *Configuring Graceful Restart for QFabric Systems*
- graceful-restart on page 392

export (Protocols BGP)

Syntax export [*policy-names*];

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Apply one or more policies to routes being exported from the routing table into BGP.

If you specify more than one policy, they are evaluated in the order specified, from left to right, and the first matching filter is applied to the route. If no routes match the filters, the routing table exports into BGP only the routes that it learned from BGP. If an action specified in one of the policies manipulates a route characteristic, the policy framework software carries the new route characteristic forward during the evaluation of the remaining policies. For example, if the action specified in the first policy of a chain sets a route's metric to 500, this route matches the criterion of **metric 500** defined in the next policy.

Options ***policy-names***—Name of one or more policies.

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

Related Documentation

- Understanding Route Advertisement on page 173
- *Routing Policy Feature Guide for Routing Devices*
- import on page 400

family (Protocols BGP)

Syntax

```
family {
  (inet | inet6 | inet-vpn | inet6-vpn | iso-vpn) {
    (any | flow | labeled-unicast | multicast | unicast) {
      accepted-prefix-limit {
        maximum number;
        teardown <percentage-threshold> idle-timeout (forever | minutes);
      }
      add-path {
        send {
          path-count number;
          prefix-policy [ policy-names ];
          }
        receive;
      }
      aigp [disable];
      loops number;
      prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
      }
      protection;
    rib-group group-name;
      topology name {
        community {
          target identifier;
        }
      }
    flow {
      no-validate policy-name;
    }
    labeled-unicast {
      accepted-prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
      }
      aggregate-label {
        community community-name:
      }
      explicit-null {
        connected-only;
      }
      prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
      }
      resolve-vpn;
      rib (inet.3 | inet6.3);
      rib-group group-name;
      traffic-statistics {
        file filename <world-readable | no-world-readable>;
        interval seconds;
      }
```

```
    }
  }
  route-target {
    accepted-prefix-limit {
      maximum number;
      proxy-generate <route-target-policy route-target-policy-name>;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
    advertise-default;
    external-paths number;
    prefix-limit {
      maximum number;
      teardown <percentage> <idle-timeout (forever | minutes)>;
    }
  }
  (evpn | inet-mdt | inet-mvpn | inet6-mvpn | l2vpn) {
    signaling {
      accepted-prefix-limit {
        maximum number;
        teardown <percentage-threshold> idle-timeout (forever | minutes);
      }
      add-path {
        send {
          path-count number;
          prefix-policy [ policy-names ];
          }
        receive;
      }
      aigp [disable];
      damping;
      loops number;
      prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
      }
      rib-group group-name;
    }
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
inet-mvpn and **inet6-mpvn** statements introduced in Junos OS Release 8.4.
inet-mdt statement introduced in Junos OS Release 9.4.
Support for the **loops** statement introduced in Junos OS Release 9.6.
evpn statement introduced in Junos OS Release 13.2.

Description

Enable multiprotocol BGP (MP-BGP) by configuring BGP to carry network layer reachability information (NLRI) for address families other than unicast IPv4, to specify MP-BGP to carry NLRI for the IPv6 address family, or to carry NLRI for VPNs.

Options

any—Configure the family type to be both unicast and multicast.

evpn—Configure NLRI parameters for Ethernet VPNs (EVPNs).

inet—Configure NLRI parameters for IPv4.

inet6—Configure NLRI parameters for IPv6.

inet-mdt—Configure NLRI parameters for the multicast distribution tree (MDT) subaddress family identifier (SAFI) for IPv4 traffic in Layer 3 VPNs.

inet-mvpn—Configure NLRI parameters for IPv4 for multicast VPNs.

inet6-mvpn—Configure NLRI parameters for IPv6 for multicast VPNs.

inet-vpn—Configure NLRI parameters for IPv4 for Layer 3 VPNs.

inet6-vpn—Configure NLRI parameters for IPv6 for Layer 3 VPNs.

iso-vpn—Configure NLRI parameters for IS-IS for Layer 3 VPNs.

l2vpn—Configure NLRI parameters for IPv4 for MPLS-based Layer 2 VPNs and VPLS.

labeled-unicast—Configure the family type to be labeled-unicast. This means that the BGP peers are being used only to carry the unicast routes that are being used by labeled-unicast for resolving the labeled-unicast routes. This statement is supported only with **inet** and **inet6**.

multicast—Configure the family type to be multicast. This means that the BGP peers are being used only to carry the unicast routes that are being used by multicast for resolving the multicast routes.

unicast—Configure the family type to be unicast. This means that the BGP peers only carry the unicast routes that are being used for unicast forwarding purposes. The default family type is **unicast**.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- *Configuring IBGP Sessions Between PE Routers in VPNs*
- *Understanding Multiprotocol BGP*
- *autonomous-system*
- local-as on page 409

graceful-restart (Protocols BGP)

Syntax

```
graceful-restart {
  disable;
  restart-time seconds;
  stale-routes-time seconds;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*
neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Configure graceful restart for BGP. Graceful restart allows a routing device undergoing a restart to inform its adjacent neighbors and peers of its condition. Graceful restart is disabled by default.

To configure the duration of the BGP graceful restart period, include the **restart-time** statement at the **[edit protocols bgp graceful-restart]** hierarchy level. To set the length of time the router waits to receive messages from restarting neighbors before declaring them down, include the **stale-routes-time** statement at the **[edit protocols bgp graceful-restart]** hierarchy level.

NOTE: If you configure graceful restart after a BGP session has been established, the BGP session restarts and the peers negotiate graceful restart capabilities.

Configure graceful restart globally at the **[edit routing-options]** or **[edit routing-instances *instance-name* routing-options]** hierarchy level to enable the feature. You cannot enable graceful restart for specific protocols unless graceful restart is also enabled globally. You can, optionally, modify the global settings at the individual protocol level.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- *Configuring Graceful Restart Options for BGP*
- *Configuring Graceful Restart for QFabric Systems*
- *Junos OS High Availability Library for Routing Devices*

group (Protocols BGP)

Syntax

```
group group-name {
  advertise-inactive;
  allow [ network/mask-length ];
  authentication-key key;
  cluster cluster-identifier;
  damping;
  description text-description;
  export [ policy-names ];
  family {
    (inet | inet6 | inet-vpn | inet6-vpn | l2-vpn) {
      (any | multicast | unicast | signaling) {
        accepted-prefix-limit {
          maximum number;
          teardown <percentage> <idle-timeout (forever | minutes)>;
        }
        add-path {
          send {
            path-count number;
            prefix-policy [ policy-names ];
            }
          receive;
        }
        aigp [disable];
        damping;
        prefix-limit {
          maximum number;
          teardown <percentage> <idle-timeout (forever | minutes)>;
        }
        rib-group group-name;
        topology name {
          community {
            target identifier;
          }
        }
      }
      flow {
        no-validate policy-name;
      }
      labeled-unicast {
        accepted-prefix-limit {
          maximum number;
          teardown <percentage> <idle-timeout (forever | minutes)>;
        }
        explicit-null {
          connected-only;
        }
        prefix-limit {
          maximum number;
          teardown <percentage> <idle-timeout (forever | minutes)>;
        }
        resolve-vpn;
        rib inet.3;
```

```
            rib-group group-name;
          }
        }
        route-target {
          accepted-prefix-limit {
            maximum number;
            teardown <percentage> <idle-timeout (forever | minutes)>;
          }
          advertise-default;
          external-paths number;
          prefix-limit {
            maximum number;
            teardown <percentage> <idle-timeout (forever | minutes)>;
          }
        }
      }
      hold-time seconds;
      import [ policy-names ];
      ipsec-sa ipsec-sa;
      keep (all | none);
      local-address address;
      local-as autonomous-system <private>;
      local-preference local-preference;
      log-updown;
      metric-out metric;
      multihop <ttl-value>;
      multipath {
        multiple-as;
      }
      no-aggregator-id;
      no-client-reflect;
      out-delay seconds;
      passive;
      peer-as autonomous-system;
      preference preference;
      remove-private;
      tcp-mss segment-size;
      traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
      }
      type type;
      neighbor address {
        ... peer-specific-options ...
      }
    }
```

Hierarchy Level

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

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Define a BGP peer group. BGP peer groups share a common type, peer autonomous system (AS) number, and cluster ID, if present. To configure multiple BGP groups, include multiple **group** statements.

By default, the group's options are identical to the global BGP options. To override the global options, include group-specific options within the **group** statement.

The **group** statement is one of the statements you must include in the configuration to run BGP on the routing device.

Each group must contain at least one peer.

Options ***group-name***—Name of the BGP group.

The remaining statements are explained separately.

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

Related Documentation
- *BGP Feature Guide for Routing Devices*

hold-down-interval (BGP BFD Liveness Detection)

Syntax holddown-interval *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit protocols bgp bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]

Release Information
Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure an interval specifying how long a BFD session must remain up before a state change notification is sent.

When you configure the hold-down interval for the BFD protocol for EBGP, the BFD session is unaware of the BGP session during this time. In this case, if the BGP session goes down during the configured hold-down interval, BFD already assumes the BGP session is down and does not send a state change notification. The **holddown-interval** statement is supported only for EBGP peers at the **[edit protocols bgp group *group-name* neighbor *address*]** hierarchy level. If the BFD session goes down and then comes back up during the configured hold-down interval, the timer is restarted. You must configure the hold-down interval on both EBGP peers. If you configure the hold-down interval for a multihop EBGP session, you must also configure a local IP address by including the **local-address** statement at the **[edit protocols bgp group *group-name*]** hierarchy level.

Options ***milliseconds***—Specify the hold-down interval value.

Range: 0 through 255,000

Default: 0

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

Related Documentation

- *Example: Configuring BFD for Static Routes*
- bfd-liveness-detection on page 372

hold-time (Protocols BGP)

Syntax hold-time *seconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Specify the hold-time value to use when negotiating a connection with the peer. The hold-time value is advertised in open packets and indicates to the peer the length of time that it should consider the sender valid. If the peer does not receive a keepalive, update, or notification message within the specified hold time, the BGP connection to the peer is closed and routing devices through that peer become unavailable.

The hold time is three times the interval at which keepalive messages are sent.

BGP on the local routing device uses the smaller of either the local hold-time value or the peer's hold-time value received in the open message as the hold time for the BGP connection between the two peers.

Starting in Junos OS Release 12.3, the BGP hold-time value can be zero (0). This implies that the speaker does not expect keepalive messages from its peer to maintain the BGP session. When negotiating between two peers, if one side requests a nonzero hold time and the other requests a zero hold time, the negotiation settles on the nonzero value and keepalive intervals are determined accordingly. Both sides must be set to zero for keepalive messages to stop being sent.

Options ***seconds***—Hold time.

Range: 10 through 65,535 seconds (or 0 for infinite hold time)

Default: 90 seconds

TIP: When you set a hold-time value of 1 though 19 seconds, we recommend that you also configure the BGP **precision-timers** statement. The **precision-timers** statement ensures that if scheduler slip messages occur, the routing device continues to send keepalive messages. When the **precision-timers** statement is included, keepalive message generation is performed in a dedicated kernel thread, which helps to prevent BGP session flaps.

Required Privilege Level

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

Related Documentation

- BGP Messages Overview on page 7
- *precision-timers*

import (Protocols BGP)

Syntax import [*policy-names*];

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Apply one or more routing policies to routes being imported into the Junos OS routing table from BGP.

If you specify more than one policy, they are evaluated in the order specified, from left to right, and the first matching filter is applied to the route. If no match is found, BGP places into the routing table only those routes that were learned from BGP routing devices. The policy framework software evaluates the routing policies in a chain sequentially. If an action specified in one of the policies manipulates a route characteristic, the policy framework software carries the new route characteristic forward during the evaluation of the remaining policies. For example, if the action specified in the first policy of a chain sets a route's metric to 500, this route matches the criterion of **metric 500** defined in the next policy.

It is also important to understand that in Junos OS, although an import policy (inbound route filter) might reject a route, not use it for traffic forwarding, and not include it in an advertisement to other peers, the router retains these routes as hidden routes. These hidden routes are not available for policy or routing purposes. However, they do occupy memory space on the router. A service provider filtering routes to control the amount of information being kept in memory and processed by a router might want the router to entirely drop the routes being rejected by the import policy.

Hidden routes can be viewed by using the **show route receive-protocol bgp *neighbor-address* hidden** command. The hidden routes can then be retained or dropped from the routing table by configuring the **keep all | none** statement at the **[edit protocols bgp]** or **[edit protocols bgp group *group-name*]** hierarchy level.

The rules of BGP route retention are as follows:

- By default, all routes learned from BGP are retained, except those where the AS path is looped. (The AS path includes the local AS.)
- By configuring the **keep all** statement, all routes learned from BGP are retained, even those with the local AS in the AS path.
- By configuring the **keep none** statement, all routes received are discarded. When this statement is configured and the inbound policy changes, Junos OS re-advertises all the routes advertised by the peer.

Options

policy-names—Name of one or more policies.

Required Privilege Level

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

Related Documentation

- Example: Configuring BGP Interactions with IGPs on page 169
- Understanding Route Advertisement on page 173
- *Routing Policy Feature Guide for Routing Devices*
- **export on page 387**

include-mp-next-hop

Syntax include-mp-next-hop;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Enable multiprotocol updates to contain next-hop reachability information.

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

Related Documentation
- *Examples: Configuring Multiprotocol BGP*

keep

Syntax keep (all | none);

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Control whether or not Junos OS keeps in memory and hides certain routes.

If the **keep none** statement is used, Junos OS does not retain in memory and hide routes that are rejected because of a BGP import policy. Nor does BGP keep in memory and hide routes that are declared unfeasible due to BGP sanity checks. The **keep none** statement causes Junos OS to discard from memory the routes that are rejected due to BGP-specific logic or BGP evaluation. When a route is rejected because of some non-BGP-specific reason, the **keep none** statement has no effect on this route. This rejected route is retained in memory and hidden even though **keep none** is configured. An example of this type of hidden route is a route for which the protocol nexthop is unresolved.

The routing table can retain the route information learned from BGP in one of the following ways:

- Default (omit the **keep** statement)—Keep all route information that was learned from BGP, except for routes whose AS path is looped and whose loop includes the local AS.
- **keep all**—Keep all route information that was learned from BGP.
- **keep none**—Discard routes that were received from a peer and that were rejected by import policy or other sanity checking, such as AS path or next hop. When you configure **keep none** for the BGP session and the inbound policy changes, Junos OS forces readvertisement of the full set of routes advertised by the peer.

In an AS path healing situation, routes with looped paths theoretically could become usable during a soft reconfiguration when the AS path loop limit is changed. However, there is a significant memory usage difference between the default and **keep all**.

Consider the following scenarios:

- A peer readvertises routes back to the peer from which it learned them.

  This can happen in the following cases:

  - Another vendor's routing device advertises the routes back to the sending peer.
  - The Junos OS peer's default behavior of not readvertising routes back to the sending peer is overridden by configuring **advertise-peer-as**.

- A provider edge (PE) routing device discards any VPN route that does not have any of the expected route targets.

When **keep all** is configured, the behavior of discarding routes received in the above scenarios is overridden.

CAUTION: When you configure keep (all | none), the associated BGP sessions are restarted.

Default By default, BGP retains incoming rejected routes in memory and hides them. If you do not include the **keep** statement, most routes are retained in the routing table. BGP keeps all route information that was learned from BGP, except for routes whose AS path is looped and whose loop includes the local AS.

Options **all**—Retain all routes.

none—Discard routes that were received from a peer and that were rejected by import policy or other sanity checking. When **keep none** is configured for the BGP session and the inbound policy changes, Junos OS forces readvertisement of the full set of routes advertised by the peer.

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

Related Documentation

- out-delay on page 438
- *Interprovider VPN Example—MP-EBGP Between ISP Peer Routers*
- *Example: Configuring Conditional Installation of Prefixes in a Routing Table*

key-chain (BGP BFD Authentication)

Syntax key-chain *key-chain-name*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication],
[edit protocols bgp bgp bfd-liveness-detection authentication],
[edit protocols bgp group *group-name* bgp bfd-liveness-detection authentication],
[edit protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication]

Release Information
Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Associate a security key with the specified BFD session using the name of the security keychain. Each key has a unique start time within the keychain. Keychain authentication allows you to change the password information periodically without bringing down peering sessions. This keychain authentication method is referred to as *hitless* because the keys roll over from one to the next without resetting any peering sessions or interrupting the routing protocol.

Options
key-chain-name—Name of the authentication keychain. The keychain name must match one of the keychains configured with the **key-chain *key-chain-name*** statement at the **[edit security authentication-key-chain]** hierarchy level.

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

Related Documentation
- *Example: Configuring BFD for Static Routes*
- *Example: Configuring BFD Authentication for Static Routes*
- Example: Configuring BFD on Internal BGP Peer Sessions on page 216
- Example: Configuring BGP Route Authentication on page 299

- Example: Configuring EBGP Multihop Sessions on page 181

local-address (Protocols BGP)

Syntax local-address *address*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Specify the address of the local end of a BGP session. This address is used to accept incoming connections to the peer and to establish connections to the remote peer. When none of the operational interfaces are configured with the specified local address, a session with a BGP peer is placed in the idle state.

You generally configure a local address to explicitly configure the system's IP address from BGP's point of view. This IP address can be either an IPv6 or IPv4 address. Typically, an IP address is assigned to a loopback interface, and that IP address is configured here.

For internal BGP (IBGP) peering sessions, generally the loopback interface (lo0) is used to establish connections between the IBGP peers. The loopback interface is always up as long as the device is operating. If there is a route to the loopback address, the IBGP peering session stays up. If a physical interface address is used instead and that interface goes up and down, the IBGP peering session also goes up and down. Thus, the loopback interface provides fault tolerance in case the physical interface or the link goes down, if the device has link redundancy.

When a device peers with a remote device's loopback interface address, the local device expects BGP update messages to come from (be sourced by) the remote device's loopback interface address. The **local-address** statement enables you to specify the source information in BGP update messages. If you omit the **local-address** statement, the expected source of BGP update messages is based on the device's source address selection rules, which normally result in the egress interface address being the expected source of update messages. When this happens, the peering session is not established because a mismatch exists between the expected source address (the egress interface

of the peer) and the actual source (the loopback interface of the peer). To ensure that the expected source address matches the actual source address, specify the loopback interface address in the **local-address** statement.

NOTE: Although a BGP session can be established when only one of the paired routing devices has **local-address** configured, we strongly recommend that you configure **local-address** on both paired routing devices for IBGP and multihop EBGP sessions. The **local-address** statement ensures that deterministic fixed addresses are used for the BGP session end-points.

If you include the **default-address-selection** statement in the configuration, the software chooses the system default address as the source for most locally generated IP packets. For protocols in which the local address is unconstrained by the protocol specification, for example IBGP and multihop EBGP, if you do not configure a specific local address when configuring the protocol, the local address is chosen using the same methods as other locally generated IP packets.

Default If you do not configure a local address, BGP uses the routing device's source address selection rules to set the local address.

Options ***address***—IPv6 or IPv4 address of the local end of the connection.

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

Related Documentation

- Example: Configuring Internal BGP Peering Sessions on Logical Systems on page 46
- Example: Configuring Internal BGP Peer Sessions on page 35
- Understanding Internal BGP Peering Sessions on page 34
- *router-id*

local-as

Syntax local-as *autonomous-system* <loops *number*> <private | alias> <no-prepend-global-as>;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
alias option introduced in Junos OS Release 9.5.
no-prepend-global-as option introduced in Junos OS Release 9.6.

Description Specify the local autonomous system (AS) number. An AS is a set of routing devices that are under a single technical administration and generally use a single interior gateway protocol (IGP) and metrics to propagate routing information within the set of routing devices.

Internet service providers (ISPs) sometimes acquire networks that belong to a different AS. When this occur, there is no seamless method for moving the BGP peers of the acquired network to the AS of the acquiring ISP. The process of configuring the BGP peers with the new AS number can be time-consuming and cumbersome. In this case, it might not be desirable to modify peer arrangements or configuration. During this kind of transition period, it can be useful to configure BGP-enabled devices in the new AS to use the former AS number in BGP updates. This former AS number is called a *local AS*.

NOTE: If you are using BGP on the routing device, you must configure an AS number before you specify the local as number.

In Junos OS Release 9.1 and later, the AS numeric range in plain-number format is extended to provide BGP support for 4-byte AS numbers, as defined in RFC 4893, *BGP Support for Four-octet AS Number Space*.

In Junos OS Release 9.3 and later, you can also configure a 4-byte AS number using the AS-dot notation format of two integer values joined by a period: *<16-bit high-order value in decimal>.<16-bit low-order value in decimal>*. For

example, the 4-byte AS number of 65546 in plain-number format is represented as 1.10 in the AS-dot notation format.

Options

alias—(Optional) Configure the local AS as an alias of the global AS number configured for the router at the **[edit routing-options]** hierarchy level. As a result, a BGP peer considers any local AS to which it is assigned as equivalent to the primary AS number configured for the routing device. When you use the **alias** option, only the AS (global or local) used to establish the BGP session is prepended in the AS path sent to the BGP neighbor.

autonomous-system—AS number.

Range: 1 through 4,294,967,295 ($2^{32} - 1$) in plain-number format

Range: 0.0 through 65535.65535 in AS-dot notation format

loops *number*—(Optional) Specify the number of times detection of the AS number in the AS_PATH attribute causes the route to be discarded or hidden. For example, if you configure **loops 1**, the route is hidden if the AS number is detected in the path one or more times. This is the default behavior. If you configure **loops 2**, the route is hidden if the AS number is detected in the path two or more times.

NOTE: If you configure the local AS values for any BGP group, the detection of routing loops is performed using both the AS and the local AS values for all BGP groups.

If the local AS for the EBGP or IBGP peer is the same as the current AS, do not use the **local-as** statement to specify the local AS number.

When you configure the local AS within a VRF, this impacts the AS path loop-detection mechanism. All of the **local-as** statements configured on the device are part of a single AS domain. The AS path loop-detection mechanism is based on looking for a matching AS present in the domain.

Range: 1 through 10

Default: 1

no-prepend-global-as—(Optional) Specify to strip the global AS and to prepend only the local AS in AS paths sent to external peers.

private—(Optional) Configure to use the local AS only during the establishment of the BGP session with a BGP neighbor but to hide it in the AS path sent to external BGP peers. Only the global AS is included in the AS path sent to external peers.

NOTE: The **private** and **alias** options are mutually exclusive. You cannot configure both options with the same **local-as** statement.

Required Privilege Level

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

Related Documentation

- Examples: Configuring BGP Local AS on page 109
- Example: Configuring a Local AS for EBGP Sessions on page 114
- *autonomous-system*
- family on page 388

local-preference

Syntax local-preference *local-preference*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Modify the value of the **LOCAL_PREF** path attribute, which is a metric used by BGP sessions to indicate the degree of preference for an external route. The route with the highest local preference value is preferred.

The **LOCAL_PREF** path attribute always is used in inbound routing policy and is advertised to internal BGP peers and to neighboring confederations. It is never advertised to external BGP peers.

Default If you omit this statement, the **LOCAL_PREF** path attribute, if present, is not modified.

Options *local-preference*—Preference to assign to routes learned from BGP or from the group or peer.

Range: 0 through 4,294,967,295 ($2^{32} - 1$)

Default: If the **LOCAL_PREF** path attribute is present, do not modify its value. If a BGP route is received without a **LOCAL_PREF** attribute, the route is handled locally (it is stored in the routing table and advertised by BGP) as if it were received with a **LOCAL_PREF** value of 100. By default, non-BGP routes that are advertised by BGP are advertised with a **LOCAL_PREF** value of 100.

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

Related Documentation
- Example: Configuring the Local Preference Value for BGP Routes on page 57
- Understanding Internal BGP Peering Sessions on page 34

- preference on page 446

log-updown (Protocols BGP)

Syntax log-updown;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Specify to generate a log a message whenever a BGP peer makes a state transition. Messages are logged using the system logging mechanism located at the **[edit system syslog]** hierarchy level.

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

Related Documentation
- Example: Preventing BGP Session Resets on page 321
- *Junos OS Administration Library for Routing Devices*
- traceoptions on page 457

loops

Syntax loops *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp family *address-family*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family *address-family*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* local-as],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* family *address-family*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* local-as]
[edit logical-systems *logical-system-name* protocols bgp local-as],
[edit logical-systems *logical-system-name* routing-options autonomous-system *as-number*],
[edit protocols bgp family *address-family*],
[edit protocols bgp group *group-name* family *address-family*],
[edit protocols bgp group *group-name* local-as],
[edit protocols bgp group *group-name* neighbor *address* family *address-family*],
[edit protocols bgp group *group-name* neighbor *address* local-as]
[edit protocols bgp local-as],
[edit routing-options autonomous-system *as-number*]

Release Information Statement introduced in Junos OS Release 9.6.

Description Globally, for the local-AS BGP attribute, or the specified address family, allow the local device's AS number to be in the received AS paths, and specify the number of times detection of the local device's AS number in the AS_PATH attribute causes the route to be discarded or hidden. For example, if you configure **loops 1**, the route is hidden if the local device's AS number is detected in the path one or more times. This prevents routing loops and is the default behavior. If you configure **loops 2**, the route is hidden if the local device's AS number is detected in the path two or more times.

Some examples of BGP address families are as follows:

- **inet unicast**
- **inet-vpn multicast**
- **inet6 any**
- **l2vpn auto-discovery-only**
- ...

This list is truncated for brevity. For a complete list of protocol families for which you can specify the **loops** statement, enter the **help apropos loops** configuration command at the **[edit protcols bgp]** hierarchy level on your device.

```
[edit protocols bgp]
user@host# help apropos loops
set family inet unicast loops
    Allow local AS in received AS paths
set family inet unicast loops <loops>
    AS-Path loop count
set family inet multicast loops
```

```
    Allow local AS in received AS paths
set family inet multicast loops <loops>
    AS-Path loop count
set family inet flow loops
    Allow local AS in received AS paths
set family inet flow loops <loops>
    AS-Path loop count
set family inet any loops
    Allow local AS in received AS paths
set family inet any loops <loops>
    AS-Path loop count
set family inet labeled-unicast loops
    Allow local AS in received AS paths
...
```

NOTE: When you configure the **loops** statement for a specific BGP address family, that value is used to evaluate the AS path for routes received by a BGP peer for the specified address family, rather than the loops value configured for the global AS number with the **loops** statement at the **[edit routing-options autonomous-system *as-number*]** hierarchy level.

Options

number—Number of times detection of the AS number in the AS_PATH attribute causes the route to be discarded or hidden.

Range: 1 through 10

Default: 1

Required Privilege Level

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

Related Documentation

- *Example: Disabling Suppression of Route Advertisements*
- *autonomous-system*
- family on page 388
- local-as on page 409

loose-check (BGP BFD Authentication)

Syntax loose-check ;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication],
[edit protocols bgp bgp bfd-liveness-detection authentication],
[edit protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection authentication]

Release Information
Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Specify loose authentication checking on the BFD session. Use loose authentication for transitional periods only when authentication might not be configured at both ends of the BFD session.

By default, strict authentication is enabled and authentication is checked at both ends of each BFD session. Optionally, to smooth migration from nonauthenticated sessions to authenticated sessions, you can configure *loose checking*. When loose checking is configured, packets are accepted without authentication being checked at each end of the session.

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

Related Documentation

- *Example: Configuring BFD for Static Routes*
- *Example: Configuring BFD Authentication for Static Routes*
- Example: Configuring BFD on Internal BGP Peer Sessions on page 216
- Example: Configuring BGP Route Authentication on page 299
- Example: Configuring EBGP Multihop Sessions on page 181

metric-out (Protocols BGP)

Syntax metric-out (*metric* | minimum-igp *offset* | igp (delay-med-update | *offset*);

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Option **delay-med-update** introduced in Junos OS Release 9.0.

Description Specify the metric for all routes sent using the multiple exit discriminator (MED, or **MULTI_EXIT_DISC**) path attribute in update messages. This path attribute is used to discriminate among multiple exit points to a neighboring AS. If all other factors are equal, the exit point with the lowest metric is preferred.

You can specify a constant metric value by including the ***metric*** option. For configurations in which a BGP peer sends third-party next hops that require the local system to perform next-hop resolution—IBGP configurations, configurations within confederation peers, or EBGP configurations that include the **multihop** command—you can specify a variable metric by including the **minimum-igp** or **igp** option.

You can increase or decrease the variable metric calculated from the IGP metric (either from the **igp** or **minimum-igp** statement) by specifying a value for ***offset***. The metric is increased by specifying a positive value for ***offset***, and decreased by specifying a negative value for ***offset***.

In Junos OS Release 9.0 and later, you can specify that a BGP group or peer not advertise updates for the MED path attributes used to calculate IGP costs for BGP next hops unless the MED is lower. You can also configure an interval to delay when MED updates are sent by including the **med-igp-update-interval** ***minutes*** statement at the **[edit routing-options]** hierarchy level.

Options
delay-med-update—Specify that a BGP group or peer configured with the **metric-out igp** statement not advertise MED updates unless the current MED value is lower than

the previously advertised MED value, or another attribute associated with the route has changed, or the BGP peer is responding to a refresh route request.

NOTE: You cannot configure the delay-med–update statement at the global BGP level.

igp—Set the metric to the most recent metric value calculated in the IGP to get to the BGP next hop. Routes learned from an EBGP peer usually have a next hop on a directly connected interface and thus the IGP value is equal to zero. This is the value advertised.

metric—Primary metric on all routes sent to peers.

Range: 0 through 4,294,967,295 ($2^{32} - 1$)

Default: No metric is sent.

minimum-igp—Set the metric to the minimum metric value calculated in the IGP to get to the BGP next hop. If a newly calculated metric is greater than the minimum metric value, the metric value remains unchanged. If a newly calculated metric is lower, the metric value is lowered to that value. When you change a neighbor's export policy from any configuration to a configuration that sets the minimum IGP offset on an exported route, the advertised MED is not updated if the value would increase as a result, even if the previous configuration does not use a minimum IGP-based MED value. This behavior helps to prevent unnecessary route flapping when an IGP cost changes, by not forcing a route update if the metric value increases past the previous lowest known value.

offset—Increases or decreases the metric by this value.

Range: -2^{31} through $2^{31} - 1$

Default: None

Required Privilege Level

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

Related Documentation

- Example: Associating the MED Path Attribute with the IGP Metric and Delaying MED Updates on page 99
- Examples: Configuring BGP MED on page 70
- Example: Configuring the MED Attribute Directly on page 73
- Understanding the MED Attribute on page 70
- *med-igp-update-interval*

minimum-interval (BFD Liveness Detection)

Syntax minimum-interval *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection],
[edit protocols bgp bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection]

Release Information
Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.

Description Configure the minimum interval after which the local routing device transmits hello packets and then expects to receive a reply from a neighbor with which it has established a BFD session. Optionally, instead of using this statement, you can specify the minimum transmit and receive intervals separately using the minimum-interval (specified under the **transmit-interval** statement) and minimum-receive-interval statements.

Options ***milliseconds***—Specify the minimum interval value for BFD liveliness detection.

Range: 1 through 255,000

Required Privilege Level

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

Related Documentation

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- bfd-liveness-detection on page 372
- minimum-receive-interval on page 423
- transmit-interval on page 460

minimum-interval (transmit-interval)

Syntax minimum-interval *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection transmit-interval],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection transmit-interval],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection transmit-interval],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection transmit-interval],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection transmit-interval],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection transmit-interval],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection transmit-interval],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection transmit-interval],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection transmit-interval],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection transmit-interval],
[edit protocols bgp bfd-liveness-detection transmit-interval],
[edit protocols bgp group *group-name* bfd-liveness-detection transmit-interval],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection transmit-interval]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection transmit-interval]

Release Information
Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.

Description Configure the minimum interval at which the local routing device transmits hello packets to a neighbor with which it has established a BFD session. Optionally, instead of using

this statement at this hierarchy level, you can configure the minimum transmit interval using the minimum-interval statement at the **bfd-liveness-detection** hierarchy level.

Options

milliseconds—Minimum transmit interval value.

Range: 1 through 255,000

NOTE: The threshold value specified in the **threshold** statement must be greater than the value specified in the **minimum-interval** statement for the **transmit-interval** statement.

Required Privilege Level

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

Related Documentation

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- bfd-liveness-detection on page 372
- minimum-interval on page 419
- threshold on page 455

minimum-receive-interval (BFD Liveness Detection)

Syntax minimum-receive-interval *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection],
[edit protocols bgp bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection]

Release Information
Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.

Description Configure the minimum interval after which the local routing device must receive a reply from a neighbor with which it has established a BFD session. Optionally, instead of using this statement, you can configure the minimum receive interval using the **minimum-interval** statement.

Options *milliseconds*—Specify the minimum receive interval value.

Range: 1 through 255,000

Required Privilege Level

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

Related Documentation

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- bfd-liveness-detection on page 372
- minimum-interval on page 419
- transmit-interval on page 460

mtu-discovery

Syntax mtu-discovery;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure TCP path maximum transmission unit (MTU) discovery.

TCP path MTU discovery enables BGP to automatically discover the best TCP path MTU for each BGP session. In Junos OS, TCP path MTU discovery is disabled by default for all BGP neighbor sessions.

When MTU discovery is disabled, TCP sessions that are not directly connected transmit packets of 512-byte maximum segment size (MSS). These small packets minimize the chances of packet fragmentation at a device along the path to the destination. However, because most links use an MTU of at least 1500 bytes, 512-byte packets do not result in the most efficient use of link bandwidth. For directly connected EBGP sessions, MTU mismatches prevent the BGP session from being established. As a workaround, enable path MTU discovery within the EBGP group.

Path MTU discovery dynamically determines the MTU size on the network path between the source and the destination, with the goal of avoiding IP fragmentation. Path MTU discovery works by setting the Don't Fragment (DF) bit in the IP headers of outgoing packets. When a device along the path has an MTU that is smaller than the packet, the device drops the packet. The device also sends back an ICMP Fragmentation Needed (Type 3, Code 4) message that contains the device's MTU, thus allowing the source to reduce its path MTU appropriately. The process repeats until the MTU is small enough to traverse the entire path without fragmentation.

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

Related Documentation

- Example: Limiting TCP Segment Size for BGP on page 314
- *Configuring the Junos OS for IPv6 Path MTU Discovery*
- *Configuring the Junos OS for Path MTU Discovery on Outgoing GRE Tunnel Connections*

multihop

Syntax

```
multihop {
    no-nexthop-change;
    ttl ttl-value;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure an EBGP multihop session.

For Layer 3 VPNs, you configure the EBGP multihop session between the PE and CE routing devices. This allows you to configure one or more routing devices between the PE and CE routing devices.

An external confederation peer is a special case that allows unconnected third-party next hops. You do not need to configure multihop sessions explicitly in this particular case because multihop behavior is implied.

If you have external BGP confederation peer-to-loopback addresses, you still need the multihop configuration.

NOTE: You cannot configure the **accept-remote-nexthop** statement at the same time.

Default

If you omit this statement, all EBGP peers are assumed to be directly connected (that is, you are establishing a nonmultihop, or "regular," BGP session), and the default time-to-live (TTL) value is 1.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- Example: Configuring EBGP Multihop Sessions on page 181
- *Configuring EBGP Multihop Sessions Between PE and CE Routers in Layer 3 VPNs*
- accept-remote-nexthop on page 359
- *no-nexthop-change*
- *ttl*

multiplier (BFD Liveness Detection)

Syntax multiplier *number*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection],
[edit protocols bgp bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection]

Release Information
Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.

Description Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down.

Options ***number***—Number of hello packets.

Range: 1 through 255

Default: 3

Required Privilege Level

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

Related Documentation

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- bfd-liveness-detection on page 372

neighbor (Protocols BGP)

Syntax

```
neighbor address {
    accept-remote-nexthop;
    advertise-external <conditional>;
    advertise-inactive;
    (advertise-peer-as | no-advertise-peer-as);
    as-override;
    authentication-algorithm algorithm;
    authentication-key key;
    authentication-key-chain key-chain;
    cluster cluster-identifier;
    damping;
    description text-description;
    export [ policy-names ];
    family {
        (inet | inet6 | inet-mvpn | inet6-mpvn | inet-vpn | inet6-vpn | iso-vpn | l2-vpn) {
            (any | flow | multicast | unicast | signaling) {
                accepted-prefix-limit {
                    maximum number;
                    teardown <percentage> <idle-timeout (forever | minutes)>;
                }
                damping;
                prefix-limit {
                    maximum number;
                    teardown <percentage> <idle-timeout (forever | minutes)>;
                }
                rib-group group-name;
                topology name {
                    community {
                        target identifier;
                    }
                }
            }
            flow {
                no-validate policy-name;
            }
            labeled-unicast {
                accepted-prefix-limit {
                    maximum number;
                    teardown <percentage> <idle-timeout (forever | minutes)>;
                }
                aggregate-label {
                    community community-name:
                }
                explicit-null {
                    connected-only;
                }
                prefix-limit {
                    maximum number;
                    teardown <percentage> <idle-timeout (forever | minutes)>;
                }
                resolve-vpn;
                rib inet.3;
```

```
        rib-group group-name;
        topology name {
          community {
            target identifier;
          }
        }
      }
    }
    route-target {
      advertise-default;
      external-paths number;
      accepted-prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
      }
      prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
      }
    }
    signaling {
      prefix-limit {
        maximum number;
        teardown <percentage> <idle-timeout (forever | minutes)>;
      }
    }
  }
  graceful-restart {
    disable;
    restart-time seconds;
    stale-routes-time seconds;
  }
  hold-time seconds;
  import [ policy-names ];
  ipsec-sa ipsec-sa;
  keep (all | none);
  local-address address;
  local-as autonomous-system <private>;
  local-interface interface-name;
  local-preference preference;
  log-updown;
  metric-out (metric | minimum-igp <offset> | igp <offset>);
  mtu-discovery;
  multihop <ttl-value>;
  multipath {
    multiple-as;
  }
  no-aggregator-id;
  no-client-reflect;
  out-delay seconds;
  passive;
  peer-as autonomous-system;
  preference preference;
  tcp-mss segment-size;
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
```

```
        flag flag <flag-modifier> <disable>;
    }
    vpn-apply-export;
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Explicitly configure a neighbor (peer). To configure multiple BGP peers, include multiple **neighbor** statements.

By default, the peer's options are identical to those of the group. You can override these options by including peer-specific option statements within the **neighbor** statement.

The **neighbor** statement is one of the statements you can include in the configuration to define a minimal BGP configuration on the routing device. (You can include an **allow all** statement in place of a **neighbor** statement.)

Options

address—IPv6 or IPv4 address of a single peer.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- *BGP Feature Guide for Routing Devices*

no-adaptation (BFD Liveness Detection)

Syntax no-adaptation;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection],
[edit protocols bgp bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection]

Release Information
Statement introduced in Junos OS Release 9.0
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.

Description Configure BFD sessions not to adapt to changing network conditions. We recommend that you *do not* disable BFD adaptation unless it is preferable to have BFD adaptation disabled in your network.

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

Related Documentation
- *Configuring BFD for Layer 2 VPN and VPLS*

- *Example: Configuring BFD for Static Routes*
- bfd-liveness-detection on page 372

no advertise-peer-as

Syntax no-advertise-peer-as;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Enable the default behavior of suppressing AS routes.

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

Related Documentation
- Example: Configuring BGP Route Advertisement on page 173
- Understanding Route Advertisement on page 173
- advertise-peer-as on page 363

no-aggregator-id

Syntax no-aggregator-id;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Prevent different routing devices within an AS from creating aggregate routes that contain different AS paths.

Junos OS performs route aggregation, which is the process of combining the characteristics of different routes so that only a single route is advertised. Aggregation reduces the amount of information that BGP must store and exchange with other BGP systems. When aggregation occurs, the local routing device adds the local AS number and the router ID to the aggregator path attiribute. The **no-aggregator-id** statement causes Junos OS to place a 0 in the router ID field and thus eliminate the possibility of having multiple aggregate advertisements in the network, each with different path information.

Default If you omit this statement, the router ID is included in the BGP aggregator path attribute.

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

Related Documentation
- Update Messages on page 7

no-client-reflect

Syntax no-client-reflect;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Disable intracluster route redistribution by the system acting as the route reflector. Include this statement when the client cluster is fully meshed to prevent the sending of redundant route advertisements. Route reflection provides a way to decrease BGP control traffic and minimizing the number of update messages sent within the AS.

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

Related Documentation
- Example: Configuring BGP Route Reflectors on page 275
- cluster on page 379

out-delay

Syntax out-delay *seconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Control how often BGP and the routing table exchange route information by specifying how long a route must be present in the Junos OS routing table before it is exported to BGP. Use this time delay to help bundle routing updates and to avoid sending updates too often.

Alternatively or in addition, external BGP (EBGP) sessions can also use the route-flap damping mechanism upon the reception of BGP messages coming from an external neighbor.

BGP stores the route information it receives from update messages in the routing table, and the routing table exports active routes from the routing table into BGP. BGP then advertises the exported routes to its peers. The **out-delay** statement enables a form of rate limiting. The delay is added to each update for each prefix individually. When a routing device changes its best path to a destination prefix, the device does not inform its peer about the change unless the route has been present in its routing table for the specified **out-delay**. If you use **out-delay** to perform rate-limiting, you can expect a less bursty pattern of updates. You will see a pattern in which updates arrive in a steady flow, and two updates for the same prefix are always spaced by at least the **out-delay** timer value (for example, 30 seconds). Thus, the **out-delay** setting is useful for limiting oscillation (sometimes called *churn*) in a network. Keep in mind that, regardless of the **out-delay** setting, BGP peers exchange routes immediately after neighbor establishment. The **out-delay** setting is only designed to delay the exchange of routes between BGP and the local routing table.

Caution is warranted because an **out-delay** can delay convergence. If your network is configured in a way that avoids oscillation, setting an **out-delay** is not necessary.

When configured, the **out-delay** value displays as **Outbound Timer** when using **show bgp group** or **show bgp group neighbor** commands.

Default By default, the exchange of route information between BGP and the routing table occurs immediately after the routes are received. This immediate exchange of route information might cause instabilities in the network reachability information. If you omit this statement, routes are exported to BGP immediately after they have been added to the routing table.

Options *seconds*—Output delay time.

Range: 0 through 65,535 seconds

Default: 0 seconds

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

Related Documentation

- keep on page 403

outbound-route-filter

Syntax

```
outbound-route-filter {
  bgp-orf-cisco-mode;
  prefix-based {
    accept {
      (inet | inet6);
    }
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information

Statement introduced in Junos OS Release 9.2.
Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure a BGP peer to accept outbound route filters from a remote peer.

Options

accept—Specify that outbound route filters from a BGP peer be accepted.

inet—Specify that IPv4 prefix-based outbound route filters be accepted.

inet6—Specify that IPv6 prefix-based outbound route filters be accepted.

NOTE: You can specify that both IPv4 and IPv6 outbound route filters be accepted.

prefix-based—Specify that prefix-based filters be accepted.

The **bgp-orf-cisco-mode** statement is explained separately.

Required Privilege Level

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

Related Documentation
- Example: Configuring BGP Prefix-Based Outbound Route Filtering on page 177

passive (Protocols BGP)

Syntax
passive;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description
Configure the routing device so that active open messages are not sent to the peer. Once you configure the routing device to be passive, the routing device will wait for the peer to issue an open request before a message is sent.

Default
If you omit this statement, all explicitly configured peers are active, and each peer periodically sends open requests until its peer responds.

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

Related Documentation
- Example: Preventing BGP Session Flaps When VPN Families Are Configured on page 321

path-selection

Syntax

```
path-selection {
  (always-compare-med | cisco-non-deterministic | external-router-id);
  as-path-ignore;
  l2vpn-use-bgp-rules;
  med-plus-igp {
    igp-multiplier number;
    med-multiplier number;
  }
}
```

Hierarchy Level

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

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
med-plus-igp option introduced in Junos OS Release 8.1.
as-path-ignore and **l2vpn-use-bgp-rules** options introduced in Junos OS Release 10.2.

Description

Configure BGP path selection.

Default

If the **path-selection** statement is not included in the configuration, only the multiple exit discriminators (MEDs) of routes that have the same peer ASs are compared.

Options

always-compare-med—Always compare MEDs whether or not the peer ASs of the compared routes are the same.

NOTE: We recommend that you configure the **always-compare-med** option.

as-path-ignore—In the best-path algorithm, skip the step that compares the autonomous system (AS) path lengths. By default, the best-path algorithm evaluates the length of the AS paths and prefers the route with the shortest AS path length.

NOTE: The **as-path-ignore** statement is not supported with routing instances.

cisco-non-deterministic—Emulate the Cisco IOS default behavior. 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.

external-router-id—Compare the router ID between external BGP paths to determine the active path.

igp-multiplier ***number***—The multiplier value for the IGP cost to a next-hop address. This option is useful for making the MED and IGP cost comparable.

Range: 1 through 1000

Default: 1

med-multiplier ***number***—The multiplier value for the MED calculation. This option is useful for making the MED and IGP cost comparable.

Range: 1 through 1000

Default: 1

med-plus-igp—Add the IGP cost to the indirect next-hop destination to the MED before comparing MED values for path selection. This statement only affects best-path selection. It does not affect the advertised MED.

The other option is explained separately.

Required Privilege Level

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

Related Documentation

- Understanding BGP Path Selection on page 197
- Example: Ignoring the AS Path Attribute When Selecting the Best Path on page 200

peer-as (Protocols BGP)

Syntax peer-as *autonomous-system*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Specify the neighbor (peer) autonomous system (AS) number.

For EBGP, the peer is in another AS, so the AS number you specify in the **peer-as** statement must be different from the local router's AS number, which you specify in the **autonomous-system** statement. For IBGP, the peer is in the same AS, so the two AS numbers that you specify in the **autonomous-system** and **peer-as** statements must be the same.

The AS numeric range in plain-number format has been extended in Junos OS Release 9.1 to provide BGP support for 4-byte AS numbers, as defined in RFC 4893, *BGP Support for Four-octet AS Number Space*. RFC 4893 introduces two new optional transitive BGP attributes, AS4_PATH and AS4_AGGREGATOR. These new attributes are used to propagate 4-byte AS path information across BGP speakers that do not support 4-byte AS numbers. RFC 4893 also introduces a reserved, well-known, 2-byte AS number, AS 23456. This reserved AS number is called AS_TRANS in RFC 4893. All releases of the Junos OS support 2-byte AS numbers.

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

With the introduction of 4-byte AS numbers, you might have a combination of routers that support 4-byte AS numbers and 2-byte AS numbers. For more information about what happens when establishing BGP peer relationships between 4-byte and 2-byte capable routers, see the following topics:

- *Using 4-Byte Autonomous System Numbers in BGP Networks Technology Overview*.

Options

autonomous-system—AS number.

Range: 1 through 4,294,967,295 ($2^{32} - 1$) in plain-number format for 4-byte AS numbers

Range: 1 through 65,535 in plain-number format for 2-byte AS numbers (this is a subset of the 4-byte range)

Range: 0.0 through 65535.65535 in AS-dot notation format for 4-byte AS numbers

Required Privilege Level

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

Related Documentation

preference (Protocols BGP)

Syntax preference *preference*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Specify the preference for routes learned from BGP.

At the BGP global level, the preference statement sets the preference for routes learned from BGP. You can override this preference in a BGP group or peer preference statement.

At the group or peer level, the preference statement sets the preference for routes learned from the group or peer. Use this statement to override the preference set in the BGP global preference statement when you want to favor routes from one group or peer over those of another.

Options ***preference***—Preference to assign to routes learned from BGP or from the group or peer.

Range: 0 through 4,294,967,295 ($2^{32} - 1$)

Default: 170 for the primary preference

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

Related Documentation
- local-preference on page 412
- Example: Configuring the Preference Value for BGP Routes on page 191

remove-private

Syntax remove-private all replace nearest;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description When advertising AS paths to remote systems, have the local system strip private AS numbers from the AS path. The numbers are stripped from the AS path starting at the left end of the AS path (the end where AS paths have been most recently added). The routing device stops searching for private ASs when it finds the first nonprivate AS or a peer's private AS. If the AS path contains the AS number of the external BGP (EBGP) neighbor, BGP does not remove the private AS number.

NOTE: As of Junos OS 10.0R2 and higher, if there is a need to send prefixes to an EBGP peer that has an AS number that matches an AS number in the AS path, consider using the **as-override** statement instead of the **remove-private** statement.

The operation takes place after any confederation member ASs have already been removed from the AS path, if applicable.

The Junos OS recognizes the set of AS numbers that is considered private, a range that is defined in the Internet Assigned Numbers Authority (IANA) assigned numbers document.

The set of reserved AS numbers is in the range from 64,512 through 65,535.

Options **all**—Remove all private AS numbers from the original path. Do not stop the process of removing private AS numbers, even if a public AS number is encountered.

nearest—When you use the **all** and **replace** options, choose the last (right-most) public AS number encountered in the original AS path for the replacement value, as the AS path is processed from left to right. If no public AS number is encountered, the default replacement value is used. (See the **replace** option for information about the default replacement value.)

replace—When you use the **all** option, instead of a removing private AS numbers, perform a replace operation. The default replacement value for the private AS number is the local AS number at the BGP group level for the BGP peer. If you are unsure about the replacement value, check the local AS value displayed in the output of the **show bgp group *group-name*** command.

Required Privilege Level

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

Related Documentation

- Example: Removing Private AS Numbers from AS Paths on page 209

restart-time (BGP Graceful Restart)

Syntax restart-time *seconds*;

Hierarchy Level
[edit protocols (bgp | rip | ripng) graceful-restart],
[edit logical-systems *logical-system-name* protocols (bgp | rip | ripng) graceful-restart (Enabling Globally)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp graceful-restart],
[edit routing-instances *routing-instance-name* protocols bgp graceful-restart]

Release Information
Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure the duration of the BGP, RIP, or next-generation RIP (RIPng) graceful restart period.

Options ***seconds***—Length of time for the graceful restart period.

Range: 1 through 600 seconds

Default: Varies by protocol:

- BGP—120 seconds
- RIP and RIPng—60 seconds

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

Related Documentation

- *Configuring Graceful Restart Options for BGP*
- *Configuring Graceful Restart Options for RIP and RIPng*
- *Configuring Graceful Restart for QFabric Systems*
- stale-routes-time on page 451

session-mode

Syntax session-mode (automatic | multihop | single-hop);

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]

Release Information
Statement introduced in Junos OS Release 11.1.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description
Configure BFD session mode to be single-hop or multihop. By default, BGP uses single-hop BFD sessions if the peer is directly connected to the router's interface. BGP uses multihop BFD sessions if the peer is not directly connected to the router's interface. If the peer session's **local-address** option is configured, the directly connected check is based partly on the source address that would be used for BGP and BFD.

For backward compatibility, you can override the default behavior by configuring the **single-hop** or **multihop** option. Before Junos OS Release 11.1, the behavior was to assume that IBGP peer sessions were multihop.

Options
automatic—Configure BGP to use single-hop BFD sessions if the peer is directly connected to the router's interface, and multihop BFD sessions if the peer is not directly connected to the router's interface

multihop—Configure BGP to use multihop BFD sessions.

single-hop—Configure BGP to use single-hop BFD sessions.

Default: automatic

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

Related Documentation
- Example: Configuring BFD Authentication for BGP on page 226
- Example: Configuring BFD on Internal BGP Peer Sessions on page 216

- Example: Configuring BFD Authentication for BGP on page 226
- Understanding BFD Authentication for BGP on page 224

stale-routes-time

Syntax stale-routes-time *seconds*;

Hierarchy Level [edit logical-systems *logical-routing-name* protocols bgp graceful-restart],
[edit logical-systems *logical-routing-name* routing-instances *routing-instance-name* protocols bgp graceful-restart],
[edit protocols bgp graceful-restart],
[edit routing-instances *routing-instance-name* protocols bgp graceful-restart]

Release Information Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Specify the maximum time that stale routes are kept during a restart. The **stale-routes-time** statement allows you to set the length of time the routing device waits to receive messages from restarting neighbors before declaring them down.

Options ***seconds***—Time the router device waits to receive messages from restarting neighbors before declaring them down.

Range: 1 through 600 seconds

Default: 300 seconds

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

Related Documentation

- *Configuring Graceful Restart Options for BGP*
- *Configuring Graceful Restart for QFabric Systems*
- restart-time (BGP Graceful Restart) on page 449

tcp-mss (Protocols BGP)

Syntax tcp-mss *segment-size*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *neighbor-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *neighbor-name*],
[edit protocols bgp],
[edit protocol bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *neighbor-name*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *neighbor-name*]

Release Information
Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure the maximum segment size (MSS) for the TCP connection for BGP neighbors.

The MSS is only valid in increments of 2 KB. The value used is based on the value set, but is rounded down to the nearest multiple of 2048.

Options ***segment-size***—MSS for the TCP connection.

Range: 1 through 4096

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

Related Documentation
- Example: Limiting TCP Segment Size for BGP on page 314

threshold (detection-time)

Syntax threshold *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection detection-time],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection detection-time],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection detection-time],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection detection-time],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection detection-time],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection detection-time],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection detection-time],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection detection-time],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection detection-time],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection detection-time],
[edit protocols bgp bfd-liveness-detection detection-time],
[edit protocols bgp group *group-name* bfd-liveness-detection detection-time],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection detection-time],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection detection-time],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection detection-time],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection detection-time]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection detection-time],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection detection-time],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection detection-time],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection detection-time]

Release Information
Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPNs and VPLS.

Description Specify the threshold for the adaptation of the BFD session detection time. When the detection time adapts to a value equal to or greater than the threshold, a single trap and a single system log message are sent.

NOTE: The threshold value must be equal to or greater than the transmit interval.

The threshold time must be equal to or greater than the value specified in the **minimum-interval** or the **minimum-receive-interval** statement.

Options

milliseconds—Value for the detection time adaptation threshold.

Range: 1 through 255,000

Required Privilege Level

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

Related Documentation

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*

threshold (transmit-interval)

Syntax threshold *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection transmit-interval],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection transmit-interval],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection transmit-interval],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection transmit-interval],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection transmit-interval],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection transmit-interval],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection transmit-interval],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection transmit-interval],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection transmit-interval],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection transmit-interval],
[edit protocols bgp bfd-liveness-detection transmit-interval],
[edit protocols bgp group *group-name* bfd-liveness-detection transmit-interval],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection transmit-interval]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection transmit-interval]

Release Information
Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.

Description Specify the threshold for the adaptation of the BFD session transmit interval. When the transmit interval adapts to a value greater than the threshold, a single trap and a single system message are sent.

Options ***milliseconds***—Value for the transmit interval adaptation threshold.
Range: 0 through 4,294,967,295 ($2^{32} - 1$)

NOTE: The threshold value specified in the **threshold** statement must be greater than the value specified in the **minimum-interval** statement for the **transmit-interval** statement.

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

Related Documentation

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- bfd-liveness-detection on page 372

traceoptions (Protocols BGP)

Syntax

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*],
[edit protocols bgp],
[edit protocols bgp group *group-name*],
[edit protocols bgp group *group-name* neighbor *address*],
[edit routing-instances *routing-instance-name* protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name*],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
4byte-as statement introduced in Junos OS Release 9.2.
4byte-as statement introduced in Junos OS Release 9.2 for EX Series switches.

Description

Configure BGP protocol-level tracing options. To specify more than one tracing operation, include multiple flag statements.

NOTE: The **traceoptions** statement is not supported on QFabric systems.

Default

The default BGP protocol-level tracing options are inherited from the routing protocols **traceoptions** statement included at the **[edit routing-options]** hierarchy level. The default group-level trace options are inherited from the BGP protocol-level **traceoptions** statement. The default peer-level trace options are inherited from the group-level **traceoptions** statement.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file ***name***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. We recommend that you place BGP tracing output in the file **bgp-log**.

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

Range: 2 through 1000 files

Default: 10 files

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

BGP Tracing Flags

- **4byte-as**—4-byte AS events.
- **bfd**—BFD protocol events.
- **damping**—Damping operations.
- **graceful-restart**—Graceful restart events.
- **keepalive**—BGP keepalive messages. If you enable the the BGP **update** flag only, received keepalive messages do not generate a trace message.
- **nsr-synchronization**—Nonstop routing synchronization events.
- **open**—Open packets. These packets are sent between peers when they are establishing a connection.
- **packets**—All BGP protocol packets.
- **refresh**—BGP refresh packets.
- **update**—Update packets. These packets provide routing updates to BGP systems. If you enable only this flag, received keepalive messages do not generate a trace message. Use the **keepalive** flag to generate a trace message for keepalive messages.

Global Tracing Flags

- **all**—All tracing operations
- **general**—A combination of the **normal** and **route** trace operations
- **normal**—All normal operations

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **policy**—Policy operations and actions
- **route**—Routing table changes
- **state**—State transitions
- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing

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

- **detail**—Provide detailed trace information.
- **filter**—Provide filter trace information. Applies only to **route**, **damping**, and **update** tracing flags.
- **receive**—Trace the packets being received.
- **send**—Trace the packets being transmitted.

no-world-readable—(Optional) Prevent any user from reading the log file.

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

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

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

Default: 128 KB

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

Required Privilege Level

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

Related Documentation

- log-updown on page 413 statement
- *Tracing Nonstop Active Routing Synchronization Events*
- Understanding Trace Operations for BGP Protocol Traffic on page 349
- *Configuring OSPF Refresh and Flooding Reduction in Stable Topologies*

transmit-interval (BFD Liveness Detection)

Syntax

```
transmit-interval {
    minimum-interval milliseconds;
    threshold milliseconds;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection],
[edit protocols bgp bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection]

Release Information

Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.

Description

Specify the transmit interval for the **bfd-liveness-detection** statement. The negotiated transmit interval for a peer is the interval between the sending of BFD packets to peers. The receive interval for a peer is the minimum time that it requires between packets sent from its peer; the receive interval is not negotiated between peers. To determine the transmit interval, each peer compares its configured minimum transmit interval with its

peer's minimum receive interval. The larger of the two numbers is accepted as the transmit interval for that peer.

The remaining statements are explained separately.

Required Privilege Level

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

Related Documentation

- *Configuring BFD for Layer 2 VPN and VPLS*
- *Example: Configuring BFD for Static Routes*
- bfd-liveness-detection on page 372
- threshold on page 455
- minimum-interval on page 421
- minimum-receive-interval on page 423

version (BFD Liveness Detection)

Syntax version (0 | 1 | automatic);

Hierarchy Level
[edit logical-systems *logical-system-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit logical-system *logical-system-name* routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection],
[edit protocols bgp bfd-liveness-detection],
[edit protocols bgp group *group-name* bfd-liveness-detection],
[edit protocols bgp group *group-name* neighbor *address* bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address* bfd-liveness-detection]
[edit routing-instances *routing-instance-name* protocols l2vpn oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls mesh-group *mesh-group-name* neighbor *neighbor-id* oam bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols vpls oam bfd-liveness-detection]

Release Information
Statement introduced in Junos OS Release 8.1
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2 for Layer 2 VPN and VPLS.

Description Specify the BFD version for detection. You can explicitly configure BFD version 0, version 1, or the routing device can automatically detect the BFD version. By default, the routing device automatically detects the BFD version, which is either 0 or 1.

Options Configure the BFD version to detect: **0** (BFD version 0), **1** (BFD version 1), or **automatic** (autodetect the BFD version)

Default: automatic

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

Related Documentation

- *Configuring BFD for Layer 2 VPN and VPLS*
- Example: Configuring BFD Authentication for BGP on page 226
- Example: Configuring BFD on Internal BGP Peer Sessions on page 216
- Example: Configuring BFD Authentication for BGP on page 226
- Understanding BFD Authentication for BGP on page 224

PART 3

Administration

CHAPTER 12

Routine Monitoring

Monitoring BGP Routing Information

Purpose Use the monitoring functionality to monitor BGP routing information on the routing device.

Action To view BGP routing information in the CLI, enter the following commands:

- **show bgp summary**
- **show bgp neighbor**

Related Documentation

- show bgp neighbor on page 482
- show bgp summary on page 496

CHAPTER 13

Operational Commands

- clear bgp damping
- clear bgp neighbor
- clear bgp table
- show bgp group
- show bgp neighbor
- show bgp summary
- show policy damping
- show route damping

clear bgp damping

List of Syntax
Syntax on page 470
Syntax (EX Series Switch and QFX Series) on page 470

Syntax
clear bgp damping
<logical-system (all | *logical-system-name*)>
<*prefix*>

Syntax (EX Series Switch and QFX Series)
clear bgp damping
<*prefix*>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.

Description
Clear BGP route flap damping information.

Options
none—Clear all BGP route flap damping information.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

prefix—(Optional) Clear route flap damping information for only the specified destination prefix.

Required Privilege Level
clear

Related Documentation
- show policy damping on page 501
- show route damping on page 503

List of Sample Output
clear bgp damping on page 470

Output Fields
When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear bgp damping

```
user@host> clear bgp damping
```

clear bgp neighbor

List of Syntax
Syntax on page 471
Syntax (EX Series Switch and QFX Series) on page 471

Syntax

clear bgp neighbor
<as *as-number*>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<malformed-route>
<*neighbor*>
<soft | soft-inbound>
<soft-minimum-igp>

Syntax (EX Series Switch and QFX Series)

clear bgp neighbor
<as *as-number*>
<instance *instance-name*>
<malformed-route>
<*neighbor*>
<soft | soft-inbound>
<soft-minimum-igp>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
malformed-route option introduced in Junos OS Release 13.2.

Description

Perform one of the following tasks:

- Change the state of one or more BGP neighbors to **IDLE**. For neighbors in the **ESTABLISHED** state, this command drops the TCP connection to the neighbors and then reestablishes the connection.
- (**soft** keyword only) Reapply export policies or import policies, respectively, to one or more BGP neighbors without changing their state.
- (**soft-inbound** keyword only) Reapply export policies or import policies, respectively, and send refresh updates to one or more BGP neighbors without changing their state.

Options

none—Change the state of all BGP neighbors to **IDLE**.

as *as-number*—(Optional) Apply this command only to neighbors in the specified autonomous system (AS).

instance *instance-name*—(Optional) Apply this command only to neighbors for the specified routing instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

malformed-route—(Optional) Remove malformed routes. If a specific neighbor is provided, Junos OS removes malformed routes for that particular neighbor. Otherwise, Junos OS removes malformed routes for all BGP neighbors. To find routes that have

malformed attributes, run the **show route hidden** command, and look for routes marked with **MalformedAttr** in the AS path field.

neighbor—(Optional) IP address of a BGP peer. Apply this command only to the specified neighbor.

soft—(Optional) Reapply any export policies to neighbors without clearing the state.

soft-inbound—(Optional) Reapply any import policies and send refresh updates to neighbors without clearing the state.

soft-minimum-igp—(Optional) Provides soft refresh of the outbound state when the interior gateway protocol (IGP) metric is reset.

Required Privilege Level clear

Related Documentation
- show bgp neighbor on page 482

List of Sample Output clear bgp neighbor on page 472

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear bgp neighbor

```
user@host> clear bgp neighbor
```

clear bgp table

Syntax clear bgp table *table-name*
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and QFX Series) clear bgp table *table-name*

Release Information Command introduced in Junos OS Release 9.0.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.

Description Request that BGP refresh routes in a specified routing table.

Options **logical-system (all | *logical-system-name*)**—(Optional) Perform this operation on all logical systems or on a particular logical system.

table-name—Request that BGP refresh routes in the specified table.

Additional Information In some cases, a prefix limit is associated with a routing table for a VPN instance. When this limit is exceeded (for example, because of a network misconfiguration), some routes might not be inserted in the table. Such routes need to be added to the table after the network issue is resolved. Use the **clear bgp table** command to request that BGP refresh routes in a VPN instance table.

Required Privilege Level clear

List of Sample Output
clear bgp table private.inet.0 on page 473
clear bgp table inet.6 logical-system all on page 473
clear bgp table private.inet.6 logical-system ls1 on page 473
clear bgp table logical-system all inet.0 on page 473
clear bgp table logical-system ls2 private.inet.0 on page 474

Output Fields This command produces no output.

Sample Output

clear bgp table private.inet.0

```
user@host> clear bgp table private.inet.0
```

clear bgp table inet.6 logical-system all

```
user@host> clear bgp table inet.6 logical-system all
```

clear bgp table private.inet.6 logical-system ls1

```
user@host> clear bgp table private.inet.6 logical-system ls1
```

clear bgp table logical-system all inet.0

```
user@host> clear bgp table logical-system all inet.0
```

clear bgp table logical-system ls2 private.inet.0

```
user@host> clear bgp table logical-system ls2 private.inet.0
```

show bgp group

List of Syntax

Syntax on page 475
Syntax (EX Series Switch and QFX Series) on page 475

Syntax

show bgp group
<brief | detail | summary>
<*group-name*>
<exact-instance *instance-name*>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<rtf>

Syntax (EX Series Switch and QFX Series)

show bgp group
<brief | detail | summary>
<*group-name*>
<exact-instance *instance-name*>
<instance *instance-name*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
exact-instance option introduced in Junos OS Release 11.4.

Description

Display information about the configured BGP groups.

Options

none—Display group information about all BGP groups.

brief | detail | summary—(Optional) Display the specified level of output.

group-name—(Optional) Display group information for the specified group.

exact-instance *instance-name*—(Optional) Display information for the specified instance only.

instance *instance-name*—(Optional) Display information about BGP groups for all routing instances whose name begins with this string (for example, **cust1**, **cust11**, and **cust111** are all displayed when you run the **show bgp group instance cust1** command). The instance name can be master for the main instance, or any valid configured instance name or its prefix.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

rtf—(Optional) Display BGP group route targeting information.

Required Privilege Level

view

List of Sample Output

show bgp group on page 479
show bgp group brief on page 479
show bgp group detail on page 480

show bgp group rtf detail on page 481
show bgp group summary on page 481

Output Fields Table 6 on page 476 describes the output fields for the **show bgp group** command. Output fields are listed in the approximate order in which they appear.

Table 6: show bgp group Output Fields

Field Name	Field Description	Level of Output
Group Type or **Group**	Type of BGP group: **Internal** or **External**.	All levels
group-index	Index number for the BGP peer group. The index number differentiates between groups when a single BGP group is split because of different configuration options at the group and peer levels.	**rtf detail**
AS	AS number of the peer. For internal BGP (IBGP), this number is the same as **Local AS**.	**brief detail** none
Local AS	AS number of the local routing device.	**brief detail** none
Name	Name of a specific BGP group.	**brief detail** none
Index	Unique index number of a BGP group.	**brief detail** none
Flags	Flags associated with the BGP group. This field is used by Juniper Networks customer support.	**brief detail** none
Remove-private options	Options associated with the remove-private statement.	**brief detail** none
Holdtime	Maximum number of seconds allowed to elapse between successive keepalive or update messages that BGP receives from a peer in the BGP group, after which the connection to the peer is closed and routing devices through that peer become unavailable.	**brief detail** none
Export	Export policies configured for the BGP group with the **export** statement.	**brief detail** none
MED tracks IGP metric update delay	Time, in seconds, that updates to multiple exit discriminator (MED) are delayed. Also displays the time remaining before the interval is set to expire	All levels
Traffic Statistics Interval	Time between sample periods for labeled-unicast traffic statistics, in seconds.	**brief detail** none
Total peers	Total number of peers in the group.	**brief detail** none

Table 6: show bgp group Output Fields *(continued)*

Field Name	Field Description	Level of Output
Established	Number of peers in the group that are in the established state.	All levels
Active/Received/Accepted/Damped	Multipurpose field that displays information about BGP peer sessions. The field's contents depend upon whether a session is established and whether it was established in the main routing device or in a routing instance. • If a peer is not established, the field shows the state of the peer session: **Active**, **Connect**, or **Idle**. • If a BGP session is established in the main routing device, the field shows the number of active, received, accepted, and damped routes that are received from a neighbor and appear in the **inet.0** (main) and **inet.2** (multicast) routing tables. For example, **8/10/10/2 and 2/4/4/0** indicate the following: ◦ 8 active routes, 10 received routes, 10 accepted routes, and 2 damped routes from a BGP peer appear in the **inet.0** routing table. ◦ 2 active routes, 4 received routes, 4 accepted routes, and no damped routes from a BGP peer appear in the **inet.2** routing table.	**summary**
ip-addresses	List of peers who are members of the group. The address is followed by the peer's port number.	All levels
Route Queue Timer	Number of seconds until queued routes are sent. If this time has already elapsed, this field displays the number of seconds by which the updates are delayed.	**detail**
Route Queue	Number of prefixes that are queued up for sending to the peers in the group.	**detail**
inet.*number*	Number of active, received, accepted, and damped routes in the routing table. For example, **inet.0: 7/10/9/0** indicates the following: • 7 active routes, 10 received routes, 9 accepted routes, and no damped routes from a BGP peer appear in the **inet.0** routing table.	none

Table 6: show bgp group Output Fields *(continued)*

Field Name	Field Description	Level of Output
Table inet.*number*	Information about the routing table. • **Received prefixes**—Total number of prefixes from the peer, both active and inactive, that are in the routing table. • **Active prefixes**—Number of prefixes received from the peer that are active in the routing table. • **Suppressed due to damping**—Number of routes currently inactive because of damping or other reasons. These routes do not appear in the forwarding table and are not exported by routing protocols. • **Advertised prefixes**—Number of prefixes advertised to a peer. • **Received external prefixes**—Total number of prefixes from the external BGP (EBGP) peers, both active and inactive, that are in the routing table. • **Active external prefixes**—Number of prefixes received from the EBGP peers that are active in the routing table. • **Externals suppressed**—Number of routes received from EBGP peers currently inactive because of damping or other reasons. • **Received internal prefixes**—Total number of prefixes from the IBGP peers, both active and inactive, that are in the routing table. • **Active internal prefixes**—Number of prefixes received from the IBGP peers that are active in the routing table. • **Internals suppressed**—Number of routes received from IBGP peers currently inactive because of damping or other reasons. • **RIB State**—Status of the graceful restart process for this routing table: **BGP restart is complete**, **BGP restart in progress**, **VPN restart in progress**, or **VPN restart is complete**.	**detail**
Groups	Total number of groups.	All levels
Peers	Total number of peers.	All levels
External	Total number of external peers.	All levels
Internal	Total number of internal peers.	All levels
Down peers	Total number of unavailable peers.	All levels
Flaps	Total number of flaps that occurred.	All levels
Table	Name of a routing table.	**brief**, none
Tot Paths	Total number of routes.	**brief**, none
Act Paths	Number of active routes.	**brief**, none
Suppressed	Number of routes currently inactive because of damping or other reasons. These routes do not appear in the forwarding table and are not exported by routing protocols.	**brief**, none

Table 6: show bgp group Output Fields *(continued)*

Field Name	Field Description	Level of Output
History	Number of withdrawn routes stored locally to keep track of damping history.	**brief**, none
Damp State	Number of active routes with a figure of merit greater than zero, but lower than the threshold at which suppression occurs.	**brief**, none
Pending	Routes being processed by the BGP import policy.	**brief**, none
Group	Group the peer belongs to in the BGP configuration.	**detail**
Receive mask	Mask of the received target included in the advertised route.	**detail**
Entries	Number of route entries received.	**detail**
Target	Route target that is to be passed by route-target filtering. If a route advertised from the provider edge (PE) routing device matches an entry in the route-target filter, the route is passed to the peer.	**detail**
Mask	Mask which specifies that the peer receive routes with the given route target.	**detail**

Sample Output

show bgp group

```
user@host> show bgp group
 Groups: 2  Peers: 2    External: 0    Internal: 2    Down peers: 1   Flaps: 0
 Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending

 inet.0
                        0          0          0          0          0          0

 bgp.l3vpn.0
                        0          0          0          0          0          0

 bgp.rtarget.0
                        2          0          0          0          0          0
```

show bgp group brief

```
user@host> show bgp group brief
 Groups: 2  Peers: 2    External: 0    Internal: 2    Down peers: 1   Flaps: 0
 Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending

 inet.0
                        0          0          0          0          0          0

 bgp.l3vpn.0
                        0          0          0          0          0          0

 bgp.rtarget.0
                        2          0          0          0          0          0
```

show bgp group detail

```
user@host> show bgp group detail
Group Type: Internal    AS: 1                     Local AS: 1
  Name: ibgp            Index: 0                  Flags: <Export Eval>
  Holdtime: 0
  Total peers: 3        Established: 0
  22.0.0.2
  22.0.0.8
  22.0.0.5

Groups: 1  Peers: 3    External: 0    Internal: 3    Down peers: 3   Flaps: 3
  Table bgp.l3vpn.0
    Received prefixes:            0
    Accepted prefixes:            0
    Active prefixes:              0
    Suppressed due to damping:    0
    Received external prefixes:   0
    Active external prefixes:     0
    Externals suppressed:         0
    Received internal prefixes:   0
    Active internal prefixes:     0
    Internals suppressed:         0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
  Table bgp.mdt.0
    Received prefixes:            0
    Accepted prefixes:            0
    Active prefixes:              0
    Suppressed due to damping:    0
    Received external prefixes:   0
    Active external prefixes:     0
    Externals suppressed:         0
    Received internal prefixes:   0
    Active internal prefixes:     0
    Internals suppressed:         0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
  Table VPN-A.inet.0
    Received prefixes:            0
    Accepted prefixes:            0
    Active prefixes:              0
    Suppressed due to damping:    0
    Received external prefixes:   0
    Active external prefixes:     0
    Externals suppressed:         0
    Received internal prefixes:   0
    Active internal prefixes:     0
    Internals suppressed:         0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
  Table VPN-A.mdt.0
    Received prefixes:            0
    Accepted prefixes:            0
    Active prefixes:              0
    Suppressed due to damping:    0
    Received external prefixes:   0
    Active external prefixes:     0
    Externals suppressed:         0
    Received internal prefixes:   0
    Active internal prefixes:     0
```

```
    Internals suppressed:         0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
```

show bgp group rtf detail

```
user@host> show bgp group rtf detail
 Group: internal (group-index: 0)
     Receive mask: 00000002
     Table: bgp.rtarget.0                                    Entries: 2
         Target                          Mask
         100:100/64                      00000002
         200:201/64                      (Group)
 Group: internal (group-index: 1)
     Table: bgp.rtarget.0                                    Entries: 1
         Target                          Mask
         200:201/64                      (Group)
```

show bgp group summary

```
user@host> show bgp group summary
Group        Type       Peers     Established    Active/Received/Accepted/Damped
ibgp         Internal   3         0

Groups: 1  Peers: 3    External: 0    Internal: 3    Down peers: 3   Flaps: 3
  bgp.l3vpn.0      : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
  bgp.mdt.0        : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
  VPN-A.inet.0     : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
  VPN-A.mdt.0      : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
```

show bgp neighbor

List of Syntax
Syntax on page 482
Syntax (EX Series Switch and QFX Series) on page 482

Syntax

```
show bgp neighbor
<exact-instance instance-name>
<instance instance-name>
<logical-system (all | logical-system-name)>
<neighbor-address>
<orf (detail | neighbor-address)
```

Syntax (EX Series Switch and QFX Series)

```
show bgp neighbor
<instance instance-name>
<exact-instance instance-name>
<neighbor-address>
<orf (neighbor-address | detail)
```

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
orf option introduced in Junos OS Release 9.2.
exact-instance option introduced in Junos OS Release 11.4.

Description

Display information about BGP peers.

Options

none—Display information about all BGP peers.

exact-instance *instance-name*—(Optional) Display information for the specified instance only.

instance *instance-name*—(Optional) Display information about BGP peers for all routing instances whose name begins with this string (for example, **cust1**, **cust11**, and **cust111** are all displayed when you run the **show bgp neighbor instance cust1** command).

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

neighbor-address—(Optional) Display information for only the BGP peer at the specified IP address.

orf (detail | *neighbor-address*)—(Optional) Display outbound route-filtering information for all BGP peers or only for the BGP peer at the specified IP address. The default is to display brief output. Use the **detail** option to display detailed output.

Additional Information

For information about the **local-address, nlri**, **hold-time,** and **preference** statements, see the *Junos OS Routing Protocols Library for Routing Devices*.

Required Privilege Level

view

Related Documentation
- clear bgp neighbor on page 471

List of Sample Output
show bgp neighbor on page 489
show bgp neighbor (CLNS) on page 490
show bgp neighbor (Layer 2 VPN) on page 491
show bgp neighbor (Layer 3 VPN) on page 493
show bgp neighbor neighbor-address on page 493
show bgp neighbor neighbor-address on page 494
show bgp neighbor orf neighbor-address detail on page 495

Output Fields
Table 7 on page 483 describes the output fields for the **show bgp neighbor** command. Output fields are listed in the approximate order in which they appear.

Table 7: show bgp neighbor Output Fields

Field Name	Field Description
Peer	Address of the BGP neighbor. The address is followed by the neighbor port number.
AS	AS number of the peer.
Local	Address of the local routing device. The address is followed by the peer port number.
Type	Type of peer: **Internal** or **External**.
State	Current state of the BGP session: • **Active**—BGP is initiating a transport protocol connection in an attempt to connect to a peer. If the connection is successful, BGP sends an Open message. • **Connect**—BGP is waiting for the transport protocol connection to be completed. • **Established**—The BGP session has been established, and the peers are exchanging update messages. • **Idle**—This is the first stage of a connection. BGP is waiting for a Start event. • **OpenConfirm**—BGP has acknowledged receipt of an open message from the peer and is waiting to receive a keepalive or notification message. • **OpenSent**—BGP has sent an open message and is waiting to receive an open message from the peer.

Table 7: show bgp neighbor Output Fields *(continued)*

Field Name	Field Description
Flags	Internal BGP flags: • **Aggregate Label**—BGP has aggregated a set of incoming labels (labels received from the peer) into a single forwarding label. • **CleanUp**—The peer session is being shut down. • **Delete**—This peer has been deleted. • **Idled**—This peer has been permanently idled. • **ImportEval**—At the last commit operation, this peer was identified as needing to reevaluate all received routes. • **Initializing**—The peer session is initializing. • **SendRtn**—Messages are being sent to the peer. • **Sync**—This peer is synchronized with the rest of the peer group. • **TryConnect**—Another attempt is being made to connect to the peer. • **Unconfigured**—This peer is not configured. • **WriteFailed**—An attempt to write to this peer failed.
Last state	Previous state of the BGP session: • **Active**—BGP is initiating a transport protocol connection in an attempt to connect to a peer. If the connection is successful, BGP sends an Open message. • **Connect**—BGP is waiting for the transport protocol connection to be completed. • **Established**—The BGP session has been established, and the peers are exchanging update messages. • **Idle**—This is the first stage of a connection. BGP is waiting for a Start event. • **OpenConfirm**—BGP has acknowledged receipt of an open message from the peer and is waiting to receive a keepalive or notification message. • **OpenSent**—BGP has sent an open message and is waiting to receive an open message from the peer.
Last event	Last activity that occurred in the BGP session: • **Closed**—The BGP session closed. • **ConnectRetry**—The transport protocol connection failed, and BGP is trying again to connect. • **HoldTime**—The session ended because the hold timer expired. • **KeepAlive**—The local routing device sent a BGP keepalive message to the peer. • **Open**—The local routing device sent a BGP open message to the peer. • **OpenFail**—The local routing device did not receive an acknowledgment of a BGP open message from the peer. • **RecvKeepAlive**—The local routing device received a BGP keepalive message from the peer. • **RecvNotify**—The local routing device received a BGP notification message from the peer. • **RecvOpen**—The local routing device received a BGP open message from the peer. • **RecvUpdate**—The local routing device received a BGP update message from the peer. • **Start**—The peering session started. • **Stop**—The peering session stopped. • **TransportError**—A TCP error occurred.

Table 7: show bgp neighbor Output Fields *(continued)*

Field Name	Field Description
Last error	Last error that occurred in the BGP session: • **Cease**—An error occurred, such as a version mismatch, that caused the session to close. • **Finite State Machine Error**—In setting up the session, BGP received a message that it did not understand. • **Hold Time Expired**—The session's hold time expired. • **Message Header Error**—The header of a BGP message was malformed. • **Open Message Error**—A BGP open message contained an error. • **None**—No errors occurred in the BGP session. • **Update Message Error**—A BGP update message contained an error.
Export	Name of the export policy that is configured on the peer.
Import	Name of the import policy that is configured on the peer.
Options	Configured BGP options: • **AddressFamily**—Configured address family: **inet** or **inet-vpn**. • **AutheKeyChain**—Authentication key change is enabled. • **DropPathAttributes**—Certain path attributes are configured to be dropped from neighbor updates during inbound processing. • **GracefulRestart**—Graceful restart is configured. • **HoldTime**—Hold time configured with the **hold-time** statement. The hold time is three times the interval at which keepalive messages are sent. • **IgnorePathAttributes**—Certain path attributes are configured to be ignored in neighbor updates during inbound processing. • **Local Address**—Address configured with the **local-address** statement. • **Multihop**—Allow BGP connections to external peers that are not on a directly connected network. • **NLRI**—Configured MBGP state for the BGP group: **multicast**, **unicast**, or both if you have configured **nlri any**. • **Peer AS**—Configured peer autonomous system (AS). • **Preference**—Preference value configured with the **preference** statement. • **Refresh**—Configured to refresh automatically when the policy changes. • **Rib-group**—Configured routing table group.
Path-attributes dropped	Path attribute codes that are dropped from neighbor updates.
Path-attributes ignored	Path attribute codes that are ignored during neighbor updates.
Authentication key change	(appears only if the **authentication-keychain** statement has been configured) Name of the authentication keychain enabled.
Authentication algorithm	(appears only if the **authentication-algorithm** statement has been configured) Type of authentication algorithm enabled: **hmac** or **md5**.
Address families configured	Names of configured address families for the VPN.

Table 7: show bgp neighbor Output Fields *(continued)*

Field Name	Field Description
Local Address	Address of the local routing device.
Remove-private options	Options associated with the remove-private statement.
Holdtime	Hold time configured with the **hold-time** statement. The hold time is three times the interval at which keepalive messages are sent.
Flags for NLRI inet-label-unicast	Flags related to labeled-unicast: • **TrafficStatistics**—Collection of statistics for labeled-unicast traffic is enabled.
Traffic statistics	Information about labeled-unicast traffic statistics: • **Options**—Options configured for collecting statistics about labeled-unicast traffic. • **File**—Name and location of statistics log files. • **size**—Size of all the log files, in bytes. • **files**—Number of log files.
Traffic Statistics Interval	Time between sample periods for labeled-unicast traffic statistics, in seconds.
Preference	Preference value configured with the **preference** statement.
Outbound Timer	Time for which the route is available in Junos OS routing table before it is exported to BGP. This field is displayed in the output only if the **out-delay** parameter is configured to a non-zero value.
Number of flaps	Number of times the BGP session has gone down and then come back up.
Peer ID	Router identifier of the peer.
Group index	Index number for the BGP peer group. The index number differentiates between groups when a single BGP group is split because of different configuration options at the group and peer levels.
Peer index	Index that is unique within the BGP group to which the peer belongs.
Local ID	Router identifier of the local routing device.
Local Interface	Name of the interface on the local routing device.
Active holdtime	Hold time that the local routing device negotiated with the peer.
Keepalive Interval	Keepalive interval, in seconds.
BFD	Status of BFD failure detection.
Local Address	Name of directly connected interface over which direct EBGP peering is established.
NLRI for restart configured on peer	Names of address families configured for restart.

Table 7: show bgp neighbor Output Fields *(continued)*

Field Name	Field Description
NLRI advertised by peer	Address families supported by the peer: **unicast** or **multicast**.
NLRI for this session	Address families being used for this session.
Peer supports Refresh capability	Remote peer's ability to send and request full route table readvertisement (route refresh capability). For more information, see RFC 2918, *Route Refresh Capability for BGP-4*.
Restart time configured on peer	Configured time allowed for restart on the neighbor.
Stale routes from peer are kept for	When graceful restart is negotiated, the maximum time allowed to hold routes from neighbors after the BGP session has gone down.
Peer does not support Restarter functionality	Graceful restart restarter-mode is disabled on the peer.
Peer does not support Receiver functionality	Graceful restart helper-mode is disabled on the peer.
Restart time requested by this peer	Restart time requested by this neighbor during capability negotiation.
Restart flag received from the peer	When this field appears, the BGP speaker has restarted (Restarting), and this peer should not wait for the **end-of-rib** marker from the speaker before advertising routing information to the speaker.
NLRI that peer supports restart for	Neighbor supports graceful restart for this address family.
NLRI peer can save forwarding state	Neighbor supporting this address family saves all forwarding states.
NLRI that peer saved forwarding for	Neighbor saves all forwarding states for this address family.
NLRI that restart is negotiated for	Router supports graceful restart for this address family.
NLRI of received end-of-rib markers	Address families for which end-of-routing-table markers are received from the neighbor.
NLRI of all end-of-rib markers sent	Address families for which end-of-routing-table markers are sent to the neighbor.
Peer supports 4 byte AS extension (peer-as 1)	Peer understands 4-byte AS numbers in BGP messages. The peer is running Junos OS Release 9.1 or later.
NLRIs for which peer can receive multiple paths	Appears in the command output of the local router if the downstream peer is configured to receive multiple BGP routes to a single destination, instead of only receiving the active route. Possible value is **inet-unicast**.

Table 7: show bgp neighbor Output Fields *(continued)*

Field Name	Field Description
NLRIs for which peer can send multiple paths: inet-unicast	Appears in the command output of the local router if the upstream peer is configured to send multiple BGP routes to a single destination, instead of only sending the active route. Possible value is **inet-unicast**.
Table inet.*number*	Information about the routing table: • **RIB State**—BGP is in the graceful restart process for this routing table: **restart is complete** or **restart in progress**. • **Bit**—Number that represents the entry in the routing table for this peer. • **Send state**—State of the BGP group: **in sync**, **not in sync**, or **not advertising**. • **Active prefixes**—Number of prefixes received from the peer that are active in the routing table. • **Received prefixes**—Total number of prefixes from the peer, both active and inactive, that are in the routing table. • **Accepted prefixes**—Total number of prefixes from the peer that have been accepted by a routing policy. • **Suppressed due to damping**—Number of routes currently inactive because of damping or other reasons. These routes do not appear in the forwarding table and are not exported by routing protocols.
Last traffic (seconds)	Last time any traffic was received from the peer or sent to the peer, and the last time the local routing device checked.
Input messages	Messages that BGP has received from the receive socket buffer, showing the total number of messages, number of update messages, number of times a policy is changed and refreshed, and the buffer size in octets. The buffer size is 16 KB.
Output messages	Messages that BGP has written to the transmit socket buffer, showing the total number of messages, number of update messages, number of times a policy is changed and refreshed, and the buffer size in octets. The buffer size is 16 KB.
Input dropped path attributes	Information about dropped path attributes: • Code—Path attribute code. • Count—Path attribute count.
Input ignored path attributes	Information about ignored path attributes: • Code—Path attribute code. • Count—Path attribute count.
Output queue	Number of BGP packets that are queued to be transmitted to a particular neighbor for a particular routing table. Output queue **0** is for unicast NLRIs, and queue **1** is for multicast NLRIs.
Trace options	Configured tracing of BGP protocol packets and operations.
Trace file	Name of the file to receive the output of the tracing operation.

Table 7: show bgp neighbor Output Fields *(continued)*

Field Name	Field Description
Filter Updates recv	(**orf** option only) Number of outbound-route filters received for each configured address family. NOTE: The counter is cumulative. For example, the counter is increased after the remote peer either resends or clears the outbound route filtering prefix list.
Immediate	(**orf** option only) Number of route updates received with the immediate flag set. The immediate flag indicates that the BGP peer should readvertise the updated routes. NOTE: The counter is cumulative. For example, the counter is increased after the remote peer either resends or clears the outbound route filtering prefix list.
Filter	(**orf** option only) Type of prefix filter received: **prefix-based** or **extended-community**.
Received filter entries	(**orf** option only) List of received filters displayed.
seq	(**orf** option only) Numerical order assigned to this prefix entry among all the received outbound route filter prefix entries.
prefix	(**orf** option only) Address for the prefix entry that matches the filter.
minlength	(**orf** option only) Minimum prefix length, in bits, required to match this prefix.
maxlength	(**orf** option only) Maximum prefix length, in bits, required to match this prefix.
match	(**orf** option only) For this prefix match, whether to **permit** or **deny** route updates.

Sample Output

show bgp neighbor

```
user@host > show bgp neighbor
Peer: 10.255.7.250+179 AS 10   Local: 10.255.7.248+63740 AS 10
  Type: Internal    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ redist_static ]
  Options: <Preference LocalAddress PeerAS Refresh>
  Local Address: 10.255.7.248 Holdtime: 90 Preference: 170 Outbound Timer: 50
  Number of flaps: 0
  Peer ID: 10.255.7.250    Local ID: 10.255.7.248      Active Holdtime: 90
  Keepalive Interval: 30         Group index: 0    Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 10)
```

```
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              1
    Received prefixes:            1
    Accepted prefixes:            1
    Suppressed due to damping:    0
    Advertised prefixes:          1
  Last traffic (seconds): Received 9    Sent 5    Checked 5
  Input messages:  Total 36     Updates 2       Refreshes 0     Octets 718
  Output messages: Total 37     Updates 1       Refreshes 0     Octets 796
  Output Queue[0]: 0

Peer: 10.255.162.214+52193 AS 100 Local: 10.255.167.205+179 AS 100
   Type: Internal    State: Established  (route reflector client)Flags: <Sync>
   Last State: OpenConfirm   Last Event: RecvKeepAlive
   Last Error: None
   Options: <Preference LocalAddress Cluster AddressFamily Rib-group Refresh>
   Address families configured: inet-unicast inet-vpn-unicast route-target
   Local Address: 10.255.167.205 Holdtime: 90 Preference: 170
   Number of flaps: 0
   Peer ID: 10.255.162.214  Local ID: 10.255.167.205    Active Holdtime: 90
   Keepalive Interval: 30         Group index: 0    Peer index: 1
```

show bgp neighbor (CLNS)

```
user@host> show bgp neighbor
Peer: 10.245.245.1+179 AS 200  Local: 10.245.245.3+3770 AS 100
  Type: External    State: Established    Flags: <ImportEval Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Multihop Preference LocalAddress HoldTime AddressFamily PeerAS
Rib-group Refresh>
  Address families configured: iso-vpn-unicast
  Local Address: 10.245.245.3 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.245.245.1     Local ID: 10.245.245.3     Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  NLRI advertised by peer: iso-vpn-unicast
  NLRI for this session: iso-vpn-unicast
  Peer supports Refresh capability (2)
  Table bgp.isovpn.0 Bit: 10000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:              3
    Received prefixes:            3
    Suppressed due to damping:    0
    Advertised prefixes:          3
  Table aaaa.iso.0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: not advertising
    Active prefixes:              3
    Received prefixes:            3
    Suppressed due to damping:    0
  Last traffic (seconds): Received 6    Sent 5    Checked 5
  Input messages:  Total 1736   Updates 4       Refreshes 0     Octets 33385
  Output messages: Total 1738   Updates 3       Refreshes 0     Octets 33305
  Output Queue[0]: 0
  Output Queue[1]: 0
```

show bgp neighbor (Layer 2 VPN)

```
user@host> show bgp neighbor
Peer: 10.69.103.2     AS 65100 Local: 10.69.103.1     AS 65103
  Type: External    State: Active         Flags: <ImportEval>
  Last State: Idle          Last Event: Start
  Last Error: None
  Export: [ BGP-INET-import ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily   PeerAS
 Refresh>
  Address families configured: inet-unicast
  Local Address: 10.69.103.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
Peer: 10.69.104.2     AS 65100 Local: 10.69.104.1     AS 65104
  Type: External    State: Active         Flags: <ImportEval>
  Last State: Idle          Last Event: Start
  Last Error: None
  Export: [ BGP-L-import ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily PeerAS
 Refresh>
  Address families configured: inet-labeled-unicast
  Local Address: 10.69.104.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
Peer: 10.255.14.182+179 AS 69    Local: 10.255.14.176+2131 AS 69
  Type: Internal    State: Established    Flags: <ImportEval>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily
  Rib-group Refresh>
  Address families configured: inet-vpn-unicast l2vpn
  Local Address: 10.255.14.176 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.255.14.182    Local ID: 10.255.14.176    Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-vpn-unicast l2vpn
  NLRI advertised by peer: inet-vpn-unicast l2vpn
  NLRI for this session: inet-vpn-unicast l2vpn
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-vpn-unicast l2vpn
  NLRI peer can save forwarding state: inet-vpn-unicast l2vpn
  NLRI that peer saved forwarding for: inet-vpn-unicast l2vpn
  NLRI that restart is negotiated for: inet-vpn-unicast l2vpn
  NLRI of received end-of-rib markers: inet-vpn-unicast l2vpn
  Table bgp.l3vpn.0 Bit: 10000
    RIB State: BGP restart in progress
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:            10
    Received prefixes:          10
    Suppressed due to damping:  0
  Table bgp.l2vpn.0 Bit: 20000
    RIB State: BGP restart in progress
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:            1
    Received prefixes:          1
    Suppressed due to damping:  0
  Table BGP-INET.inet.0 Bit: 30000
```

```
  RIB State: BGP restart in progress
  RIB State: VPN restart in progress
  Send state: in sync
  Active prefixes:            2
  Received prefixes:          2
  Suppressed due to damping:  0
Table BGP-L.inet.0 Bit: 40000
  RIB State: BGP restart in progress
  RIB State: VPN restart in progress
  Send state: in sync
  Active prefixes:            2
  Received prefixes:          2
  Suppressed due to damping:  0
Table LDP.inet.0 Bit: 50000
  RIB State: BGP restart is complete
  RIB State: VPN restart in progress
  Send state: in sync
  Active prefixes:            1
  Received prefixes:          1
  Suppressed due to damping:  0
Table OSPF.inet.0 Bit: 60000
  RIB State: BGP restart is complete
  RIB State: VPN restart in progress
  Send state: in sync
  Active prefixes:            2
  Received prefixes:          2
  Suppressed due to damping:  0
Table RIP.inet.0 Bit: 70000
  RIB State: BGP restart is complete
  RIB State: VPN restart in progress
  Send state: in sync
  Active prefixes:            2
  Received prefixes:          2
  Suppressed due to damping:  0
Table STATIC.inet.0 Bit: 80000
  RIB State: BGP restart is complete
  RIB State: VPN restart in progress
  Send state: in sync
  Active prefixes:            1
  Received prefixes:          1
  Suppressed due to damping:  0
Table L2VPN.l2vpn.0 Bit: 90000
  RIB State: BGP restart is complete
  RIB State: VPN restart in progress
  Send state: in sync
  Active prefixes:            1
  Received prefixes:          1
  Suppressed due to damping:  0
Last traffic (seconds): Received 0    Sent 0    Checked 0
Input messages:  Total 14     Updates 13      Refreshes 0     Octets 1053
Output messages: Total 3      Updates 0       Refreshes 0     Octets 105
Output Queue[0]: 0
Output Queue[1]: 0
Output Queue[2]: 0
Output Queue[3]: 0
Output Queue[4]: 0
Output Queue[5]: 0
Output Queue[6]: 0
Output Queue[7]: 0
Output Queue[8]: 0
```

show bgp neighbor (Layer 3 VPN)

```
user@host> show bgp neighbor
Peer: 4.4.4.4+179     AS 10045 Local: 5.5.5.5+1214    AS 10045
  Type: Internal    State: Established    Flags: <ImportEval>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ match-all ] Import: [ match-all ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily
   Rib-group Refresh>
  Address families configured: inet-vpn-unicast
  Local Address: 5.5.5.5 Holdtime: 90 Preference: 170
  Flags for NLRI inet-labeled-unicast: TrafficStatistics
  Traffic Statistics: Options: all File: /var/log/bstat.log
                                         size 131072 files 10
  Traffic Statistics Interval: 60
  Number of flaps: 0
  Peer ID: 192.168.1.110    Local ID: 192.168.1.111    Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-vpn-unicast
  NLRI advertised by peer: inet-vpn-unicast
  NLRI for this session: inet-vpn-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-vpn-unicast
  NLRI peer can save forwarding state: inet-vpn-unicast
  NLRI that peer saved forwarding for: inet-vpn-unicast
  NLRI that restart is negotiated for: inet-vpn-unicast
  NLRI of received end-of-rib markers: inet-vpn-unicast
  NLRI of all end-of-rib markers sent: inet-vpn-unicast
  Table bgp.l3vpn.0 Bit: 10000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:            2
    Received prefixes:          2
    Suppressed due to damping:  0
  Table vpn-green.inet.0 Bit: 20001
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:            2
    Received prefixes:          2
    Suppressed due to damping:  0
  Last traffic (seconds): Received 15   Sent 20   Checked 20
  Input messages:  Total 40     Updates 2       Refreshes 0     Octets 856
  Output messages: Total 44     Updates 2       Refreshes 0     Octets 1066
  Output Queue[0]: 0
  Output Queue[1]: 0
  Trace options: detail packets
  Trace file: /var/log/bgpgr.log size 131072 files 10
```

show bgp neighbor neighbor-address

```
user@host> show bgp neighbor 192.168.1.111
Peer: 10.255.245.12+179 AS 35  Local: 10.255.245.13+2884 AS 35
  Type: Internal    State: Established  (route reflector client)Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
```

```
  Options: <Preference LocalAddress HoldTime Cluster AddressFamily Rib-group
  Refresh>
  Address families configured: inet-vpn-unicast inet-labeled-unicast
  Local Address: 10.255.245.13 Holdtime: 90 Preference: 170
  Flags for NLRI inet-vpn-unicast: AggregateLabel
  Flags for NLRI inet-labeled-unicast: AggregateLabel
  Number of flaps: 0
  Peer ID: 10.255.245.12    Local ID: 10.255.245.13    Active Holdtime: 90
  Keepalive Interval: 30
BFD: disabled
  NLRI advertised by peer: inet-vpn-unicast inet-labeled-unicast
  NLRI for this session: inet-vpn-unicast inet-labeled-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 300
  Stale routes from peer are kept for: 60
  Restart time requested by this peer: 300
  NLRI that peer supports restart for: inet-unicast inet6-unicast
  NLRI that restart is negotiated for: inet-unicast inet6-unicast
  NLRI of received end-of-rib markers: inet-unicast inet6-unicast
  NLRI of all end-of-rib markers sent: inet-unicast inet6-unicast
  Table inet.0 Bit: 10000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 4
    Received prefixes: 6
    Suppressed due to damping: 0
  Table inet6.0 Bit: 20000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 0
    Received prefixes: 2
    Suppressed due to damping: 0
  Last traffic (seconds): Received 3    Sent 3    Checked 3
  Input messages:  Total 9      Updates 6       Refreshes 0     Octets 403
  Output messages: Total 7      Updates 3       Refreshes 0     Octets 365
  Output Queue[0]: 0
  Output Queue[1]: 0
  Trace options: detail packets
  Trace file: /var/log/bgpgr size 131072 files 10
```

show bgp neighbor neighbor-address

```
user@host> show bgp neighbor 192.168.4.222
Peer: 192.168.4.222+4902 AS 65501 Local: 192.168.4.221+179 AS 65500
  Type: External    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Cease
  Export: [ export-policy ] Import: [ import-policy ]
  Options: <Preference HoldTime AddressFamily PeerAS PrefixLimit Refresh>
  Address families configured: inet-unicast inet-multicast
  Holdtime: 60000 Preference: 170
  Number of flaps: 4
  Last flap event: RecvUpdate
  Error: 'Cease' Sent: 5 Recv: 0
  Peer ID: 10.255.245.6     Local ID: 10.255.245.5     Active Holdtime: 60000
  Keepalive Interval: 20000      Peer index: 0
  BFD: disabled, down
  Local Interface: fxp0.0
  NLRI advertised by peer: inet-unicast inet-multicast
  NLRI for this session: inet-unicast inet-multicast
  Peer supports Refresh capability (2)
```

```
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              8
    Received prefixes:            10
    Accepted prefixes:            10
    Suppressed due to damping:    0
    Advertised prefixes:          3
  Table inet.2 Bit: 20000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 357  Sent 357  Checked 357
  Input messages:  Total 4 Updates 2  Refreshes 0  Octets 211
  Output messages: Total 4 Updates 1  Refreshes 0  Octets 147
  Output Queue[0]: 0
  Output Queue[1]: 0
  Trace options:  all
  Trace file: /var/log/bgp size 10485760 files 10
```

show bgp neighbor orf neighbor-address detail

```
user@host > show bgp neighbor orf 192.168.165.56 detail
Peer: 192.168.165.56+179 Type: External
  Group: ext1

  inet-unicast
    Filter updates recv:          1 Immediate:          1
    Filter: prefix-based  receive
      Received filter entries:
        seq 1: prefix 2.2.2.2/32: minlen 32: maxlen 32: match deny:

  inet6-unicast
    Filter updates recv:          0 Immediate:          1
    Filter: prefix-based  receive
      Received filter entries:
        *:*
```

show bgp summary

List of Syntax
Syntax on page 496
Syntax (EX Series Switch and QFX Series) on page 496

Syntax
show bgp summary
<exact-instance *instance-name*>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and QFX Series)
show bgp summary
<exact-instance *instance-name*>
<instance *instance-name*>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
exact-instance option introduced in Junos OS Release 11.4.

Description
Display BGP summary information.

Options
none—Display BGP summary information for all routing instances.

exact-instance *instance-name*—(Optional) Display information for the specified instance only.

instance *instance-name*—(Optional) Display information for all routing instances whose name begins with this string (for example, **cust1**, **cust11**, and **cust111** are all displayed when you run the **show bgp summary instance cust1** command). The instance name can be master for the main instance, or any valid configured instance name or its prefix.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

List of Sample Output
show bgp summary (When a Peer Is Not Established) on page 499
show bgp summary (When a Peer Is Established) on page 499
show bgp summary (CLNS) on page 499
show bgp summary (Layer 2 VPN) on page 499
show bgp summary (Layer 3 VPN) on page 500

Output Fields
Table 8 on page 496 describes the output fields for the **show bgp summary** command. Output fields are listed in the approximate order in which they appear.

Table 8: show bgp summary Output Fields

Field Name	Field Description
Groups	Number of BGP groups.

Table 8: show bgp summary Output Fields *(continued)*

Field Name	Field Description
Peers	Number of BGP peers.
Down peers	Number of down BGP peers.
Table	Name of routing table.
Tot Paths	Total number of paths.
Act Paths	Number of active routes.
Suppressed	Number of routes currently inactive because of damping or other reasons. These routes do not appear in the forwarding table and are not exported by routing protocols.
History	Number of withdrawn routes stored locally to keep track of damping history.
Damp State	Number of routes with a figure of merit greater than zero, but still active because the value has not reached the threshold at which suppression occurs.
Pending	Routes in process by BGP import policy.
Peer	Address of each BGP peer. Each peer has one line of output.
AS	Peer's AS number.
InPkt	Number of packets received from the peer.
OutPkt	Number of packets sent to the peer.
OutQ	Number of BGP packets that are queued to be transmitted to a particular neighbor. It normally is **0** because the queue usually is emptied quickly.
Flaps	Number of times the BGP session has gone down and then come back up.
Last Up/Down	Last time since the neighbor transitioned to or from the established state.

Table 8: show bgp summary Output Fields *(continued)*

Field Name	Field Description
State|#Active /Received/Accepted /Damped	Multipurpose field that displays information about BGP peer sessions. The field's contents depend upon whether a session is established and whether it was established on the main routing device or in a routing instance. • If a peer is not established, the field shows the state of the peer session: **Active**, **Connect**, or **Idle**. In general, the Idle state is the first stage of a connection. BGP is waiting for a Start event. A session can be idle for other reasons as well. The reason that a session is idle is sometimes displayed. For example: **Idle (Removal in progress)** or **Idle (LicenseFailure)**. • If a BGP session is established on the main routing device, the field shows the number of active, received, accepted, and damped routes that are received from a neighbor and appear in the **inet.0** (main) and **inet.2** (multicast) routing tables. For example, **8/10/10/2 and 2/4/4/0** indicate the following: ◦ 8 active routes, 10 received routes, 10 accepted routes, and 2 damped routes from a BGP peer appear in the **inet.0** routing table. ◦ 2 active routes, 4 received routes, 4 accepted routes, and no damped routes from a BGP peer appear in the **inet.2** routing table. • If a BGP session is established in a routing instance, the field indicates the established **(Establ)** state, identifies the specific routing table that receives BGP updates, and shows the number of active, received, and damped routes that are received from a neighbor. For example, **Establ VPN-AB.inet.0: 2/4/0** indicates the following: ◦ The BGP session is established. ◦ Routes are received in the **VPN-AB.inet.0** routing table. ◦ The local routing device has two active routes, four received routes, and no damped routes from a BGP peer. When a BGP session is established, the peers are exchanging update messages.

Sample Output

show bgp summary (When a Peer Is Not Established)

```
user@host> show bgp summary
Groups: 2 Peers: 4 Down peers: 1
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 6          4          0          0          0          0
Peer                 AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.0.0.3         65002         86         90       0       2       42:54 0/0/0

0/0/0
10.0.0.4         65002         90         91       0       1       42:54 0/2/0

0/0/0
10.0.0.6         65002         87         90       0       3           3 Active
10.1.12.1        65001         89         89       0       1       42:54 4/4/0

0/0/0
```

show bgp summary (When a Peer Is Established)

```
user@host> show bgp summary
Groups: 1 Peers: 3 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 6          4          0          0          0          0
Peer                 AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.0.0.2         65002      88675      88652       0       2       42:38 2/4/0

0/0/0
10.0.0.3         65002      54528      54532       0       1     2w4d22h 0/0/0

 0/0/0
10.0.0.4         65002      51597      51584       0       0     2w3d22h 2/2/0

0/0/0
```

show bgp summary (CLNS)

```
user@host> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Peer                 AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.245.245.1       200       1735       1737       0       0    14:26:12 Establ
  bgp.isovpn.0: 3/3/0
  aaaa.iso.0: 3/3/0
```

show bgp summary (Layer 2 VPN)

```
user@host> show bgp summary
Groups: 1 Peers: 5 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l2vpn.0            1          1          0          0          0          0
inet.0                 0          0          0          0          0          0
Peer                 AS      InPkt     OutPkt    OutQ   Flaps Last
Up/Dwn State|#Active/Received/Damped...
10.255.245.35   65299         72         74       0       1       19:00 Establ
  bgp.l2vpn.0: 1/1/0
  frame-vpn.l2vpn.0: 1/1/0
```

```
10.255.245.36   65299       2164       2423       0       4       19:50 Establ
  bgp.l2vpn.0: 0/0/0
  frame-vpn.l2vpn.0: 0/0/0
10.255.245.37   65299         36         37       0       4       17:07 Establ
  inet.0: 0/0/0
10.255.245.39   65299        138        168       0       6       53:48 Establ
  bgp.l2vpn.0: 0/0/0
  frame-vpn.l2vpn.0: 0/0/0
10.255.245.69   65299        134        140       0       6       53:42 Establ
  inet.0: 0/0/0
```

show bgp summary (Layer 3 VPN)

```
user@host> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State Pending
bgp.l3vpn.0            2          2          0          0          0       0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.39.1.5           2         21         22       0       0        6:26 Establ
  VPN-AB.inet.0: 1/1/0
10.255.71.15        1         19         21       0       0        6:17 Establ
  bgp.l3vpn.0: 2/2/0
  VPN-A.inet.0: 1/1/0
  VPN-AB.inet.0: 2/2/0
  VPN-B.inet.0: 1/1/0
```

show policy damping

List of Syntax
Syntax on page 501
Syntax (EX Series Switch and QFX Series) on page 501

Syntax
show policy damping
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and QFX Series)
show policy damping

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.

Description
Display information about BGP route flap damping parameters.

Options
none—Display information about BGP route flap damping parameters.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Additional Information
In the output from this command, figure-of-merit values correlate with the probability of future instability of a routing device. Routes with higher figure-of-merit values are suppressed for longer periods of time. The figure-of-merit value decays exponentially over time. A figure-of-merit value of zero is assigned to each new route. The value is increased each time the route is withdrawn or readvertised, or when one of its path attributes changes.

Required Privilege Level
view

Related Documentation
- "Configuring BGP Flap Damping Parameters" in the *Routing Policy Feature Guide for Routing Devices*
- clear bgp damping on page 470
- show route damping on page 503

List of Sample Output
show policy damping on page 502

Output Fields
Table 9 on page 502 describes the output fields for the **show policy damping** command. Output fields are listed in the approximate order in which they appear.

Table 9: show policy damping Output Fields

Field Name	Field Description
Halflife	Decay half-life, in minutes. The value represents the period during which the accumulated figure-of-merit value is reduced by half if the route remains stable. If a route has flapped, but then becomes stable, the figure-of-merit value for the route decays exponentially. For example, for a route with a figure-of-merit value of 1500, if no incidents occur, its figure-of-merit value is reduced to 750 after 15 minutes and to 375 after another 15 minutes.
Reuse merit	Figure-of-merit value below which a suppressed route can be used again. A suppressed route becomes reusable when its figure-of-merit value decays to a value below a reuse threshold, and the route once again is considered usable and can be installed in the forwarding table and exported from the routing table.
Suppress/cutoff merit	Figure-of-merit value above which a route is suppressed for use or inclusion in advertisements. When a route's figure-of-merit value reaches a particular level, called the cutoff or suppression threshold, the route is suppressed. When a route is suppressed, the routing table no longer installs the route into the forwarding table and no longer exports this route to any of the routing protocols.
Maximum suppress time	Maximum hold-down time, in minutes. The value represents the maximum time that a route can be suppressed no matter how unstable it has been before this period of stability.
Computed values	• **Merit ceiling**—Maximum merit that a flapping route can collect. • **Maximum decay**—Maximum decay half-life, in minutes.

Sample Output

show policy damping

```
user@host> show policy damping
Default damping information:
  Halflife: 15 minutes
  Reuse merit: 750 Suppress/cutoff merit: 3000
  Maximum suppress time: 60 minutes
  Computed values:
    Merit ceiling: 12110
    Maximum decay: 6193
Damping information for "standard-damping":
  Halflife: 10 minutes
  Reuse merit: 4000 Suppress/cutoff merit: 8000
  Maximum suppress time: 30 minutes
  Computed values:
    Merit ceiling: 32120
    Maximum decay: 12453
```

show route damping

List of Syntax
Syntax on page 503
Syntax (EX Series Switch and QFX Series) on page 503

Syntax
show route damping (decayed | history | suppressed)
<brief | detail | extensive | terse>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and QFX Series)
show route damping (decayed | history | suppressed)
<brief | detail | extensive | terse>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.

Description
Display the BGP routes for which updates might have been reduced because of route flap damping.

Options
brief | detail | extensive | terse—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief.

decayed—Display route damping entries that might no longer be valid, but are not suppressed.

history—Display entries that have already been withdrawn, but have been logged.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

suppressed—Display entries that have been suppressed and are no longer being installed into the forwarding table or exported by routing protocols.

Required Privilege Level
view

Related Documentation
- clear bgp damping on page 470
- show policy damping on page 501

List of Sample Output
show route damping decayed detail on page 506
show route damping history on page 507
show route damping history detail on page 507

Output Fields
Table 10 on page 504 lists the output fields for the **show route damping** command. Output fields are listed in the approximate order in which they appear.

Table 10: show route damping Output Fields

Field Name	Field Description	Level of Output
routing-table-name	Name of the routing table—for example, **inet.0**.	All levels
destinations	Number of destinations for which there are routes in the routing table.	All levels
number **routes**	Number of routes in the routing table and total number of routes in the following states: • **active** • **holddown** (routes that are in a pending state before being declared inactive) • **hidden** (the routes are not used because of a routing policy)	All levels
destination-prefix **(entry, announced)**	Destination prefix. The **entry** value is the number of routes for this destination, and the **announced** value is the number of routes being announced for this destination.	**detail extensive**
[*protocol***,** *preference***]**	Protocol from which the route was learned and the preference value for the route. • **+**—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table. • **-** —A hyphen indicates the last active route. • *****—An asterisk indicates that the route is both the active and the last active route. An asterisk before a **to** line indicates the best subpath to the route. In every routing metric except for the BGP **LocalPref** attribute, a lesser value is preferred. 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.	All levels
Next-hop reference count	Number of references made to the next hop.	**detail extensive**
Source	IP address of the route source.	**detail extensive**
Next hop	Network layer address of the directly reachable neighboring system.	**detail extensive**
via	Interface used to reach the next hop. If there is more than one interface available to the next hop, the interface that is actually used is followed by the word **Selected**.	**detail extensive**
Protocol next hop	Network layer address of the remote routing device that advertised the prefix. This address is used to derive a forwarding next hop.	**detail extensive**
Indirect next hop	Index designation used to specify the mapping between protocol next hops, tags, kernel export policy, and the forwarding next hops.	**detail extensive**
State	Flags for this route. For a description of possible values for this field, see the output field table for the *show route detail* command.	**detail extensive**

Table 10: show route damping Output Fields *(continued)*

Field Name	Field Description	Level of Output
Local AS	AS number of the local routing device.	**detail extensive**
Peer AS	AS number of the peer routing device.	**detail extensive**
Age	How long the route has been known.	**detail extensive**
Metric	Metric for the route.	**detail extensive**
Task	Name of the protocol that has added the route.	**detail extensive**
Announcement bits	List of protocols that announce this route. ***n*-Resolve inet** indicates that the route is used for route resolution for next hops found in the routing table. ***n*** is an index used by Juniper Networks customer support only.	**detail extensive**
AS path	AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated: • **I**—IGP. • **E**—EGP. • **?**—Incomplete; typically, the AS path was aggregated. When AS path numbers are included in the route, the format is as follows: • **[ ]**—Brackets enclose the local AS number associated with the AS path if more than one AS number is configured on the routing device or if AS path prepending is configured. • **{ }**—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. • **( )**—Parentheses enclose a confederation. • **([ ])**—Parentheses and brackets enclose a confederation set. NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.	All levels
to	Next hop to the destination. An angle bracket (>) indicates that the route is the selected route.	**brief** none
via	Interface used to reach the next hop. If there is more than one interface available to the next hop, the interface that is actually used is followed by the word **Selected**.	**brief** none
Communities	Community path attribute for the route. See the output field table for the *show route detail* command.	**detail extensive**
Localpref	Local preference value included in the route.	All levels
Router ID	BGP router ID as advertised by the neighbor in the open message.	**detail extensive**

Table 10: show route damping Output Fields *(continued)*

Field Name	Field Description	Level of Output
Merit (last update/now)	Last updated and current figure-of-merit value.	**detail extensive**
damping-parameters	Name that identifies the damping parameters used, which is defined in the damping statement at the **[edit policy-options]** hierarchy level.	**detail extensive**
Last update	Time of most recent change in path attributes.	**detail extensive**
First update	Time of first change in path attributes, which started the route damping process.	**detail extensive**
Flaps	Number of times the route has gone up or down or its path attributes have changed.	**detail extensive**
Suppressed	(**suppressed** keyword only) This route is currently suppressed. A suppressed route does not appear in the forwarding table and routing protocols do not export it.	All levels
Reusable in	(**suppressed** keyword only) Time when a suppressed route will again be available.	All levels
Preference will be	(**suppressed** keyword only) Preference value that will be applied to the route when it is again active.	All levels

Sample Output

show route damping decayed detail

```
user@host> show route damping decayed detail
inet.0: 173319 destinations, 1533668 routes (172625 active, 4 holddown, 108083
hidden)
10.0.111.0/24 (7 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next-hop reference count: 151973
                Source: 172.23.2.129
                Next hop: via so-1/2/0.0
                Next hop: via so-5/1/0.0, selected
                Next hop: via so-6/0/0.0
                Protocol next hop: 172.23.2.129
                Indirect next hop: 89a1a00 264185
                State: <Active Ext>
                Local AS: 65000 Peer AS:   65490
                Age: 3:28       Metric2: 0
                Task: BGP_65490.172.23.2.129+179
                Announcement bits (6): 0-KRT 1-RT 4-KRT 5-BGP.0.0.0.0+179

               6-Resolve tree 2 7-Resolve tree 3
                AS path: 65490 65520 65525 65525 65525 65525 I ()
                Communities: 65501:390 65501:2000 65501:3000 65504:701
                Localpref: 100
                Router ID: 172.23.2.129
                Merit (last update/now): 1934/1790
                damping-parameters: damping-high
```

```
                    Last update:       00:03:28 First update:       00:06:40
                    Flaps: 2
```

show route damping history

```
user@host> show route damping history
inet.0: 173320 destinations, 1533529 routes (172624 active, 6 holddown, 108122
hidden)
+ = Active Route, - = Last Active, * = Both

10.108.0.0/15       [BGP ] 2d 22:47:58, localpref 100
                      AS path: 65220 65501 65502 I
                    > to 192.168.60.85 via so-3/1/0.0
```

show route damping history detail

```
user@host> show route damping history detail
inet.0: 173319 destinations, 1533435 routes (172627 active, 2 holddown, 108105
hidden)
10.108.0.0/15 (3 entries, 1 announced)
         BGP                 /-101
                Next-hop reference count: 69058
                Source: 192.168.60.85
                Next hop: 192.168.60.85 via so-3/1/0.0, selected
                State: <Hidden Ext>
                Inactive reason: Unusable path
                Local AS: 65000 Peer AS:  65220
                Age: 2d 22:48:10
                Task: BGP_65220.192.168.60.85+179
                AS path: 65220 65501 65502 I ()
                Communities: 65501:390 65501:2000 65501:3000 65504:3561
                Localpref: 100
                Router ID: 192.168.80.25
                Merit (last update/now): 1000/932
                damping-parameters: set-normal
                Last update:       00:01:05 First update:       00:01:05
                Flaps: 1
```