JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

Multitopology Routing User Guide

Published
2020-09-21

JUNOS

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

Junos® OS Multitopology Routing User Guide

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 https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation | v

Documentation and Release Notes | v

Using the Examples in This Manual | v

- Merging a Full Example | vi
- Merging a Snippet | vii

Documentation Conventions | vii

Documentation Feedback | x

Requesting Technical Support | x

- Self-Help Online Tools and Resources | xi
- Creating a Service Request with JTAC | xi

1 **Overview**

Understanding Multitopology Routing | 13

- Routing Table Naming Conventions for Multitopology Routing | 14
- Filter-Based Forwarding Support | 15

Standards for Multitopology Routing | 16

2 **Configuring Multitopology Routing**

Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 18

Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 20

Understanding Multitopology Routing in Conjunction with PIM | 52

Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths | 55

3 **Monitoring Multitopology Routing**

Example: Tracing Global Routing Protocol Operations | 90

4 Troubleshooting Network Issues

Working with Problems on Your Network | 97

Isolating a Broken Network Connection | 97

Identifying the Symptoms of a Broken Network Connection | 99

Isolating the Causes of a Network Problem | 100

Taking Appropriate Action for Resolving the Network Problem | 101

Evaluating the Solution to Check Whether the Network Problem Is Resolved | 103

5 Configuration Statements

rib (Multitopology Routing) | 106

topologies (Multitopology Routing) | 108

topology (Filter-Based Forwarding) | 110

topology (Multitopology Routing) | 112

topology (OSPF) | 113

topology (OSPF Interface) | 115

topology (Protocols BGP) | 117

topology-id | 119

6 Operational Commands

show (ospf | ospf3) interface | 121

About the Documentation

IN THIS SECTION

- Documentation and Release Notes | v
- Using the Examples in This Manual | v
- Documentation Conventions | vii
- Documentation Feedback | x
- Requesting Technical Support | x

Use this guide to configure class-based forwarding for different types of traffic using a combination of routing protocols and firewall filters on your Juniper Networks devices.

Documentation and Release Notes

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

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

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

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

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

Documentation Conventions

Table 1 on page viii 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 viii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*

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

Convention	Description	Examples
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		

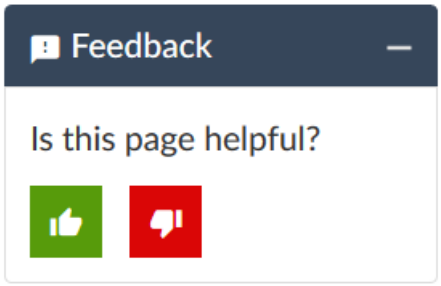
Table 2: Text and Syntax Conventions *(continued)*

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

Documentation Feedback

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

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



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active Juniper Care or Partner Support Services support contract, or are

covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Create a service request online: https://myjuniper.juniper.net

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

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

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

CHAPTER 1

Overview

Understanding Multitopology Routing | **13**

Standards for Multitopology Routing | **16**

Understanding Multitopology Routing

IN THIS SECTION

- Routing Table Naming Conventions for Multitopology Routing | 14
- Filter-Based Forwarding Support | 15

Multitopology routing enables you to configure class-based forwarding for different types of traffic, such as voice, video, and data. Each type of traffic is defined by a topology that is used to create a new routing table for that topology.

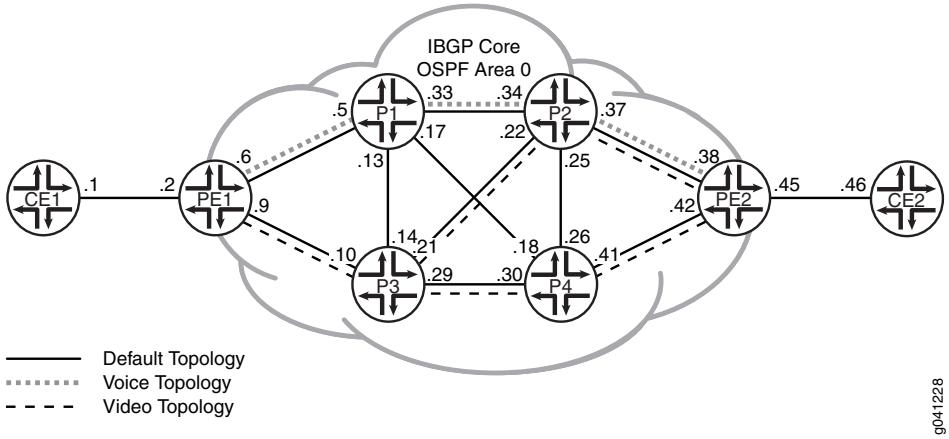
Service providers and enterprises can use multitopology routing (MTR) to engineer traffic flow across a network. MTR can be used with direct and static routes, OSPF, and BGP. MTR can be configured for unicast and multicast IP, using the Junos® operating system (Junos OS). With basic unicast IP, an IBGP core runs on top of OSPF to direct traffic based on application types, such as voice or video. For multicast, Protocol Independent Multicast (PIM) is used, in conjunction with multitopology OSPF and BGP, to direct multicast traffic over particular paths based on traffic characteristics.

In a network carrying multiple traffic types, you often need to direct different types of application traffic over multiple links depending on their link characteristics. For example, voice traffic requires links that are less likely to incur latency, jitter, or packet loss. File traffic, on the other hand, requires links that have large amounts of available bandwidth. MTR is a way to direct traffic to follow specified paths. You can use MTR to extend a traditional MPLS network into a segment where only IP forwarding is supported. With MTR, each traffic type is handled in its own conceptually incongruent topology, and yet runs on top of the same underlying network. You can configure separate topologies to share the same network links as needed. MTR uses a combination of control plane (routing) and forwarding plane (firewall filters). Each topology uses the unified control plane to make routing decisions for traffic associated with that topology. In addition, each topology has a separate forwarding table and, in effect, a dedicated forwarding plane for each topology. This forwarding plane not only directs traffic using its own forwarding table, but also simultaneously handles sophisticated functionality, such as queuing for class of service (CoS), that can be applied to a topology. As traffic enters the router, fields within a packet determine to which topology the traffic belongs.

Multitopology routing enables you to configure class-based forwarding for different types of traffic, such as voice, video, and data. Each type of traffic is defined by a topology that is used to create a new routing table for that topology. MTR provides the ability to generate forwarding tables based on the resolved entries in the routing tables for the custom topologies you create. In this way, packets of different classes can be routed independently from one another.

One way to manage traffic flow is to group links into specific routing topologies based on application requirements. Each routing topology can be thought of as of a set of contiguous links. MTR provides a way for you to manage each set of links uniquely by directing traffic types to flow over specified links. This solution uses a combination of routing (control plane) and firewall filtering (forwarding plane) configurations. Figure 1 on page 14 shows a network with two topologies configured: voice (dotted lines) and video (dashed lines). Note there is also a default routing topology (solid line).

Figure 1: Voice and Video Routing Topologies Enabled on a Subset of Links



You can configure MTR for BGP, OSPF, and static routes. When a routing topology is created, it has its own forwarding table.

Packet forwarding uses firewall filters to examine packets as they enter the router over an interface. These filters determine whether a specific topology or the default forwarding table should be used to make packet forwarding decisions. If applicable, firewall filters evaluate packet attributes, such as destination IP address, Differentiated Services code points (DSCPs), or next-level protocol headers, to determine which topology to use. In fact, any item in a packet that is recognized by a firewall filter can be used to direct the packet next-hop lookup to use a particular topology. Once the packet is directed to use a topology, the destination IP address must be in the topology forwarding table. Otherwise, the packet is dropped.

The following topics provide background information about multitopology routing:

Routing Table Naming Conventions for Multitopology Routing

Routing topologies have their own routing tables, similar to any other routing table created by default or by a **rib-group** configuration with a few differences. The routing table name indicates that the routing table is associated with a topology by prepending a colon (**:**) to the name. For example, a routing topology named voice has a routing table named **:voice.inet.0**. When routing topologies are configured under **routing-options**, a new routing table for each topology is created.

Each routing protocol creates a routing table based on the topology name, the instance name, and the purpose of the table. A routing table for each topology uses the following format:

logical-system-name/routing-instance-name*:*topology-name*.*protocol*.*identifier

The routing instance string is included only if the instance is not the master. The logical system string is included only if the logical system identifier has a value other than 0 (zero). Each routing table for a topology includes a colon (:) before the topology name that also separates the routing-instance name from the topology name. ***protocol*** is the protocol family, which can be **inet** or **inet6**. ***identifier*** is a positive integer that specifies the instance of the routing table. Table 3 on page 15 shows specific examples of routing tables for various topologies.

Table 3: Examples of Routing Tables for Custom Topologies

Name of Routing Table	Description
:voice.inet.0	Master instance, voice topology, unicast IPv4 routes
:voice.inet6.0	Master instance, voice topology, unicast IPv6 routes
:voice.inet.3	Master instance, voice topology, ingress label-switched paths (LSPs)
private_1/:voice.inet.0	Logical system private, voice topology, unicast IPv4 routes
customer-A:voice.inet.0	Virtual-router customer-A, voice topology, unicast IPv4 routes
customer-B:voice.inet.3	Virtual-router customer-B, voice topology, ingress LSPs
customer-A:voice.mpls.0	Virtual-router customer-A, voice topology, unicast carrier-of-carriers IPv4 routes

To run multitopology routing (MTR), you must configure IP routing. MTR supports OSPF version 2 (OSPFv2), static routes, and BGP. You must configure an interior gateway protocol (IGP), such as OSPFv2 or static routing. Configure BGP to add routes learned through BGP to the appropriate custom topologies.

MTR is also supported on logical systems and the virtual router routing instance. No other routing instance type is supported on MTR.

Filter-Based Forwarding Support

By default, the ingress interface forwards traffic to the default topology for each configured routing instance. MTR supports filter-based forwarding, which enables you to match traffic on the ingress interface with a specific type of forwarding class and then forward that traffic to the specified topology. You can

further define how traffic is handled for each forwarding class by configuring additional firewall filters that match traffic for such values as the IP precedence field or the Differentiated Services code point (DSCP).

RELATED DOCUMENTATION

- Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | **18**
- Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | **20**
- Understanding Multitopology Routing in Conjunction with PIM | **52**
- Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths | **55**

Standards for Multitopology Routing

Multitopology routing is defined in the following document:

- RFC 4915, *Multi-Topology (MT) Routing in OSPF*

2

CHAPTER

Configuring Multitopology Routing

Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | **18**

Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | **20**

Understanding Multitopology Routing in Conjunction with PIM | **52**

Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths | **55**

Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic

Multitopology routing (MTR) enables you to configure class-based forwarding for different types of traffic, such as voice, video, and data. Each type of traffic is defined by a topology that is used to create a new routing table for that topology. MTR provides the ability to generate forwarding tables based on the resolved entries in the routing tables for the custom topologies you create. In this way, packets of different classes can be routed independently from one another.

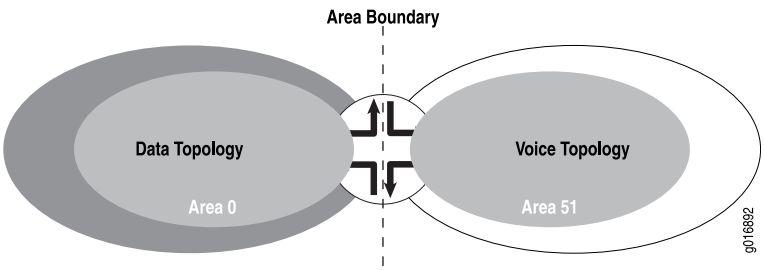
To run MTR, you must configure IP routing. MTR supports OSPFv2, static routes, and BGP. You must configure an interior gateway protocol (IGP), such as OSPFv2 or static routing, Configure BGP to add routes learned through BGP to the appropriate custom topologies. MTR also supports filter-based forwarding, which enables you to match traffic on the ingress interface with a specific type of forwarding class and then forward that traffic to the specified topology.

OSPF in MTR

OSPF in MTR uses a single instance of OSPF to carry connectivity and IP reachability information for different topologies. That information is used to calculate shortest-path-first (SPF) trees and routing tables. OSPF for MTR supports protocol extensions that include metrics that correspond to different topologies for link and prefix reachabilty information. The type-of-service (TOS) metric field is used to advertise the topology-specific metric for links and prefixes belonging to that topology. The TOS field is redefined as MT-ID in the payload of router, summary, and Type 5 and Type 7 AS-external link-state advertisements (LSAs).

Under MTR, each OSPF interface continues to belong to a single area. Therefore, by default, all topologies share the same area boundaries. As a result, attributes of an area, such as stubbiness, are independent of the topology. By default, all topologies configured for OSPF are enabled on all interfaces. However, you can disable one or more configured topologies on an interface. You can thus allocate an interface for a specific topology. In Figure 2 on page 18, Area 51 includes an interface that is uniquely allocated to voice traffic, and Area 0 includes an interface that is uniquely allocated to data traffic. Each topology thus corresponds to a different OSPF area that shares a boundary.

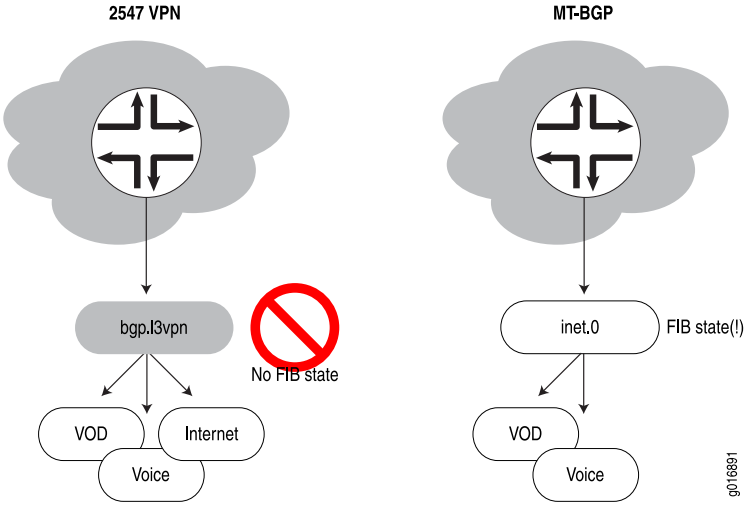
Figure 2: MTR-OSPF Area Boundary



BGP in MTR

BGP in MTR provides the ability to resolve BGP routes against configured topologies. An inbound policy is used to select routes for inclusion in the appropriate routing tables for the topologies. The default behavior for virtual private networks (VPNs) that use MPLS for forwarding packets over the backbone and that use BGP for distributing routes over the backbone is to place BGP route updates in the **bgp.l3vpn** routing table. Figure 3 on page 19 shows a BGP peer operating in an environment that conforms with the requirements in RFC 2547, *BGP/MPLS VPNs*. The figure shows how a BGP peer configured for MTR performs secondary route resolution.

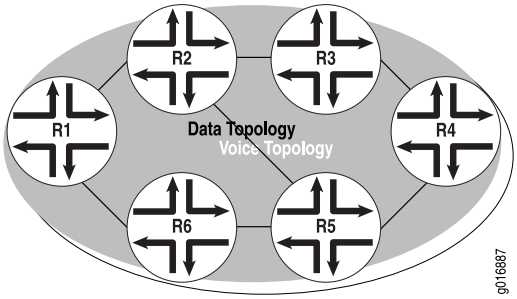
Figure 3: BGP Route Resolution



The BGP peer in a standard VPN topology places prefixes for routes it learns in the **bgp.l3vpn** routing table, which does not result in automatic updates to the forwarding table. Under BGP in MTR, when BGP receives a route from a peer, it attempts to resolve that route against a route in the **inet.0** routing table. If the route is resolved, it is placed in that table, which generates a forwarding state. If you have configured a community target identifier that matches the import policy for the topology, routing and forwarding states are added to the tables for the topology.

Because MTR provides support for BGP to perform secondary route resolution, as Figure 4 on page 20 shows, MTR is able to create two distinct network paths for each type of traffic. Each router advertises BGP routes that need to be resolved against the IGP routes for each topology. Based on the IGP metrics configured for each topology, for all routes that originate from Router 4 (R4), the upper path between R1 and R4, which traverses R2 and R3, is selected for voice traffic, whereas the lower path between R1 and R4, which traverses R5 and R6, is selected for data traffic.

Figure 4: Route Resolution for MTR



RELATED DOCUMENTATION

Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 20

Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic

IN THIS SECTION

- Requirements | 21
- Overview | 21
- Configuration | 22
- Verification | 43

This example shows how to use multitopology routing (MTR) to choose a topology path based on an application, either voice or video.

Requirements

This example requires that Junos OS Release 9.0 or later is running on the provider core devices.

Overview

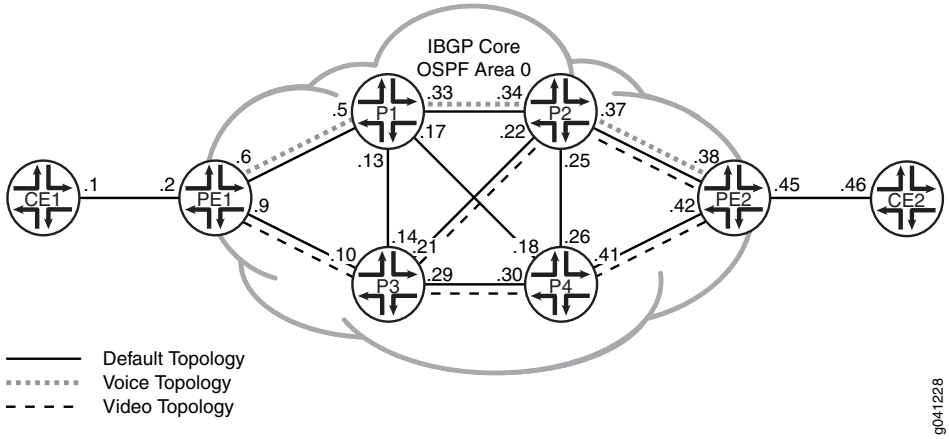
In this example, the network is running OSPF and internal BGP (IBGP) in the core, but not MPLS. Even without traffic engineering, voice traffic uses one set of links, and video traffic uses a different set of links. This traffic might or might not be destined for the same IP address. In some cases, both applications traverse the same link. The solution uses MTR-based OSPF and BGP, along with firewall filters to direct different traffic types over designated links. The routers use a fairly similar set of configurations, which reduces complexity and improves network management.

The OSPF topologies are defined to support each service offering over the OSPF area. The links of a topology must be contiguous, consistent with a typical OSPF area. IBGP routes in each routing topology automatically use the associated OSPF topology routing table for protocol next-hop route resolution. No special route resolution configurations are required. In this solution, multiple topologies can be configured over the same link. However, traffic in each application service class cannot traverse links unless they are configured for the topology designated for that service. Figure 5 on page 21 shows a diagram of this case. Contiguous paths for routing the voice topology are shown with dotted lines, and paths for routing the video topology are shown with dashed lines.

For a complete set of configurations for all of the devices in the topology, see "CLI Quick Configuration" on page 22. The remainder of the example focuses on Device CE1 and Device PE1.

Figure 5 on page 21 shows the sample topology.

Figure 5: Multitopology OSPF and IBGP for Designating Links Belonging to Voice and Video Services



Configuration

IN THIS SECTION

- Configuring Device CE1 | 32
- Configuring Device PE1 | 35

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 CE1

```
set interfaces fe-0/1/0 fastether-options loopback
set interfaces fe-0/1/0 unit 0 family inet address 11.19.130.1/24
set interfaces fe-0/1/0 unit 0 family inet address 11.19.131.1/24
set interfaces fe-0/1/0 unit 0 family inet address 11.19.132.1/24
set interfaces fe-1/2/0 unit 1 description to-PE1
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 97 family inet address 10.255.165.97/32 primary
set protocols bgp group ebgp type external
set protocols bgp group ebgp local-address 10.0.0.1
set protocols bgp group ebgp export set_community
set protocols bgp group ebgp export inject_directs
set protocols bgp group ebgp peer-as 100
set protocols bgp group ebgp neighbor 10.0.0.2
set policy-options policy-statement inject_directs term a from protocol direct
set policy-options policy-statement inject_directs term a from interface fe-0/1/0.0
set policy-options policy-statement inject_directs term a then next policy
set policy-options policy-statement inject_directs term a then accept
set policy-options policy-statement inject_directs term b then reject
set policy-options policy-statement set_community term a from route-filter 11.19.130.0/24 exact
set policy-options policy-statement set_community term a from route-filter 11.19.131.0/24 exact
set policy-options policy-statement set_community term a then community add voice
set policy-options policy-statement set_community term a then accept
set policy-options policy-statement set_community term b from route-filter 11.19.132.0/24 exact
set policy-options policy-statement set_community term b from route-filter 11.19.133.0/24 exact
set policy-options policy-statement set_community term b then community add video
```

```
set policy-options policy-statement set_community term b then accept
set policy-options policy-statement set_community term default then accept
set policy-options community video members target:50:50
set policy-options community voice members target:40:40
set routing-options autonomous-system 101
```

Device CE2

```
set interfaces fe-0/1/1 fastether-options loopback
set interfaces fe-0/1/1 unit 0 family inet address 11.19.140.1/24
set interfaces fe-0/1/1 unit 0 family inet address 11.19.141.1/24
set interfaces fe-0/1/1 unit 0 family inet address 11.19.142.1/24
set interfaces fe-1/2/0 unit 46 description to-PE2
set interfaces fe-1/2/0 unit 46 family inet address 10.0.0.46/30
set interfaces lo0 unit 20 family inet address 10.255.165.20/32 primary
set protocols bgp group ebgp type external
set protocols bgp group ebgp local-address 10.0.0.46
set protocols bgp group ebgp export set_community
set protocols bgp group ebgp export inject_directs
set protocols bgp group ebgp peer-as 100
set protocols bgp group ebgp neighbor 10.0.0.45
set policy-options policy-statement inject_directs term a from protocol direct
set policy-options policy-statement inject_directs term a from interface fe-0/1/1.0
set policy-options policy-statement inject_directs term a then next policy
set policy-options policy-statement inject_directs term a then accept
set policy-options policy-statement inject_directs term b then reject
set policy-options policy-statement set_community term a from route-filter 11.19.140.0/24 exact
set policy-options policy-statement set_community term a from route-filter 11.19.141.0/24 exact
set policy-options policy-statement set_community term a then community add voice
set policy-options policy-statement set_community term a then accept
set policy-options policy-statement set_community term b from route-filter 11.19.142.0/24 exact
set policy-options policy-statement set_community term b from route-filter 11.19.143.0/24 exact
set policy-options policy-statement set_community term b then community add video
set policy-options policy-statement set_community term b then accept
set policy-options policy-statement set_community term default then accept
set policy-options community video members target:50:50
set policy-options community voice members target:40:40
set routing-options autonomous-system 102
```

Device PE1

```
set interfaces fe-1/2/0 unit 2 description to-CE1
set interfaces fe-1/2/0 unit 2 family inet filter input ef_path
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 6 description to-P1
set interfaces fe-1/2/1 unit 6 family inet filter input ef_path
set interfaces fe-1/2/1 unit 6 family inet address 10.0.0.6/30
set interfaces fe-1/2/2 unit 9 description to-P3
set interfaces fe-1/2/2 unit 9 family inet filter input ef_path
set interfaces fe-1/2/2 unit 9 family inet address 10.0.0.9/30
set interfaces lo0 unit 93 family inet address 10.255.165.93/32 primary
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.93
set protocols bgp group ibgp family inet unicast topology voice community target:40:40
set protocols bgp group ibgp family inet unicast topology video community target:50:50
set protocols bgp group ibgp export nhs
set protocols bgp group ibgp neighbor 10.255.165.111
set protocols bgp group ibgp neighbor 10.255.165.203
set protocols bgp group ibgp neighbor 10.255.165.113
set protocols bgp group ibgp neighbor 10.255.165.95
set protocols bgp group ibgp neighbor 10.255.165.99
set protocols bgp group ebgp type external
set protocols bgp group ebgp local-address 10.0.0.2
set protocols bgp group ebgp family inet unicast topology voice community target:40:40
set protocols bgp group ebgp family inet unicast topology video community target:50:50
set protocols bgp group ebgp peer-as 101
set protocols bgp group ebgp neighbor 10.0.0.1
set protocols ospf topology voice topology-id 126
set protocols ospf topology video topology-id 52
set protocols ospf area 0.0.0.0 interface fe-1/2/1.6 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/1.6 topology video disable
set protocols ospf area 0.0.0.0 interface fe-1/2/1.6 topology voice
set protocols ospf area 0.0.0.0 interface fe-1/2/2.9 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/2.9 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/2.9 topology video
set protocols ospf area 0.0.0.0 interface lo0.93 passive
set policy-options policy-statement nhs then next-hop self
set routing-options autonomous-system 100
set routing-options topologies family inet topology voice
set routing-options topologies family inet topology video
set firewall family inet filter ef_path term ef from forwarding-class expedited-forwarding
set firewall family inet filter ef_path term ef then topology voice
set firewall family inet filter ef_path term video from source-address 11.19.132.0/24
```

```
set firewall family inet filter ef_path term video from source-address 11.19.133.0/24
set firewall family inet filter ef_path term video from source-address 11.19.142.0/24
set firewall family inet filter ef_path term video from source-address 11.19.144.0/24
set firewall family inet filter ef_path term video then topology video
set firewall family inet filter ef_path term default then accept
set class-of-service interfaces fe-1/2/0 unit 2 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/1 unit 6 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/2 unit 9 classifiers inet-precedence default
```

Device PE2

```
set interfaces fe-1/2/0 unit 38 description to-P2
set interfaces fe-1/2/0 unit 38 family inet filter input ef_path
set interfaces fe-1/2/0 unit 38 family inet address 10.0.0.38/30
set interfaces fe-1/2/1 unit 42 description to-P4
set interfaces fe-1/2/1 unit 42 family inet filter input ef_path
set interfaces fe-1/2/1 unit 42 family inet address 10.0.0.42/30
set interfaces fe-1/2/2 unit 45 description to-CE2
set interfaces fe-1/2/2 unit 45 family inet filter input ef_path
set interfaces fe-1/2/2 unit 45 family inet address 10.0.0.45/30
set interfaces lo0 unit 203 family inet address 10.255.165.203/32 primary
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.203
set protocols bgp group ibgp family inet unicast topology voice community target:40:40
set protocols bgp group ibgp family inet unicast topology video community target:50:50
set protocols bgp group ibgp export nhs
set protocols bgp group ibgp neighbor 10.255.165.111
set protocols bgp group ibgp neighbor 10.255.165.93
set protocols bgp group ibgp neighbor 10.255.165.113
set protocols bgp group ibgp neighbor 10.255.165.95
set protocols bgp group ibgp neighbor 10.255.165.99
set protocols bgp group ebgp type external
set protocols bgp group ebgp local-address 10.0.0.45
set protocols bgp group ebgp family inet unicast topology voice community target:40:40
set protocols bgp group ebgp family inet unicast topology video community target:50:50
set protocols bgp group ebgp peer-as 102
set protocols bgp group ebgp neighbor 10.0.0.46
set protocols ospf topology voice topology-id 126
set protocols ospf topology video topology-id 52
set protocols ospf area 0.0.0.0 interface fe-1/2/0.38 metric 10
```

```
set protocols ospf area 0.0.0.0 interface fe-1/2/0.38 topology video metric 200
set protocols ospf area 0.0.0.0 interface fe-1/2/0.38 topology voice
set protocols ospf area 0.0.0.0 interface fe-1/2/1.42 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/1.42 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/1.42 topology video
set protocols ospf area 0.0.0.0 interface lo0.203 passive
set policy-options policy-statement nhs then next-hop self
set routing-options autonomous-system 100
set routing-options topologies family inet topology voice
set routing-options topologies family inet topology video
set firewall family inet filter ef_path term ef from forwarding-class expedited-forwarding
set firewall family inet filter ef_path term ef then topology voice
set firewall family inet filter ef_path term video from source-address 11.19.132.0/24
set firewall family inet filter ef_path term video from source-address 11.19.133.0/24
set firewall family inet filter ef_path term video from source-address 11.19.142.0/24
set firewall family inet filter ef_path term video from source-address 11.19.144.0/24
set firewall family inet filter ef_path term video then topology video
set firewall family inet filter ef_path term default then accept
set class-of-service interfaces fe-1/2/0 unit 38 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/1 unit 42 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/2 unit 45 classifiers inet-precedence default
```

Device P1

```
set interfaces fe-1/2/0 unit 5 description to-PE1
set interfaces fe-1/2/0 unit 5 family inet filter input ef_path
set interfaces fe-1/2/0 unit 5 family inet address 10.0.0.5/30
set interfaces fe-1/2/1 unit 13 description to-P3
set interfaces fe-1/2/1 unit 13 family inet filter input ef_path
set interfaces fe-1/2/1 unit 13 family inet address 10.0.0.13/30
set interfaces fe-1/2/2 unit 17 description to-P4
set interfaces fe-1/2/2 unit 17 family inet filter input ef_path
set interfaces fe-1/2/2 unit 17 family inet address 10.0.0.17/30
set interfaces fe-1/2/3 unit 33 description to-P2
set interfaces fe-1/2/3 unit 33 family inet filter input ef_path
set interfaces fe-1/2/3 unit 33 family inet address 10.0.0.33/30
set interfaces lo0 unit 99 family inet address 10.255.165.99/32 primary
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.99
set protocols bgp group ibgp family inet unicast topology voice community target:40:40
```

```
set protocols bgp group ibgp family inet unicast topology video community target:50:50
set protocols bgp group ibgp neighbor 10.255.165.93
set protocols bgp group ibgp neighbor 10.255.165.111
set protocols bgp group ibgp neighbor 10.255.165.203
set protocols bgp group ibgp neighbor 10.255.165.95
set protocols bgp group ibgp neighbor 10.255.165.113
set protocols ospf topology voice topology-id 126
set protocols ospf topology video topology-id 52
set protocols ospf area 0.0.0.0 interface fe-1/2/3.33 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/3.33 topology voice
set protocols ospf area 0.0.0.0 interface fe-1/2/3.33 topology video disable
set protocols ospf area 0.0.0.0 interface fe-1/2/2.17 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/2.17 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/2.17 topology video disable
set protocols ospf area 0.0.0.0 interface fe-1/2/1.13 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/1.13 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/1.13 topology video disable
set protocols ospf area 0.0.0.0 interface fe-1/2/0.5 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/0.5 topology voice
set protocols ospf area 0.0.0.0 interface fe-1/2/0.5 topology video disable
set protocols ospf area 0.0.0.0 interface lo0.99 passive
set routing-options autonomous-system 100
set routing-options topologies family inet topology voice
set routing-options topologies family inet topology video
set firewall family inet filter ef_path term ef from forwarding-class expedited-forwarding
set firewall family inet filter ef_path term ef then topology voice
set firewall family inet filter ef_path term video from source-address 11.19.132.0/24
set firewall family inet filter ef_path term video from source-address 11.19.133.0/24
set firewall family inet filter ef_path term video from source-address 11.19.142.0/24
set firewall family inet filter ef_path term video from source-address 11.19.144.0/24
set firewall family inet filter ef_path term video then topology video
set firewall family inet filter ef_path term default then accept
set class-of-service interfaces fe-1/2/0 unit 5 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/1 unit 13 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/2 unit 17 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/3 unit 33 classifiers inet-precedence default
```

Device P2

```
set interfaces fe-1/2/0 unit 22 description to-P3
```

```
set interfaces fe-1/2/0 unit 22 family inet filter input ef_path
set interfaces fe-1/2/0 unit 22 family inet address 10.0.0.22/30
set interfaces fe-1/2/1 unit 25 description to-P4
set interfaces fe-1/2/1 unit 25 family inet filter input ef_path
set interfaces fe-1/2/1 unit 25 family inet address 10.0.0.25/30
set interfaces fe-1/2/2 unit 34 description to-P1
set interfaces fe-1/2/2 unit 34 family inet filter input ef_path
set interfaces fe-1/2/2 unit 34 family inet address 10.0.0.34/30
set interfaces fe-1/2/3 unit 37 description to-PE2
set interfaces fe-1/2/3 unit 37 family inet filter input ef_path
set interfaces fe-1/2/3 unit 37 family inet address 10.0.0.37/30
set interfaces lo0 unit 113 family inet address 10.255.165.113/32 primary
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.113
set protocols bgp group ibgp family inet unicast topology voice community target:40:40
set protocols bgp group ibgp family inet unicast topology video community target:50:50
set protocols bgp group ibgp neighbor 10.255.165.93
set protocols bgp group ibgp neighbor 10.255.165.111
set protocols bgp group ibgp neighbor 10.255.165.203
set protocols bgp group ibgp neighbor 10.255.165.95
set protocols bgp group ibgp neighbor 10.255.165.99
set protocols ospf topology voice topology-id 126
set protocols ospf topology video topology-id 52
set protocols ospf area 0.0.0.0 interface fe-1/2/2.34 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/2.34 topology voice
set protocols ospf area 0.0.0.0 interface fe-1/2/2.34 topology video disable
set protocols ospf area 0.0.0.0 interface fe-1/2/0.22 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/0.22 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/0.22 topology video metric 20
set protocols ospf area 0.0.0.0 interface fe-1/2/1.25 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/1.25 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/1.25 topology video disable
set protocols ospf area 0.0.0.0 interface fe-1/2/3.37 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/3.37 topology voice
set protocols ospf area 0.0.0.0 interface fe-1/2/3.37 topology video metric 200
set protocols ospf area 0.0.0.0 interface lo0.113 passive
set routing-options autonomous-system 100
set routing-options topologies family inet topology voice
set routing-options topologies family inet topology video
set firewall family inet filter ef_path term ef from forwarding-class expedited-forwarding
set firewall family inet filter ef_path term ef then topology voice
set firewall family inet filter ef_path term video from source-address 11.19.132.0/24
```

```
set firewall family inet filter ef_path term video from source-address 11.19.133.0/24
set firewall family inet filter ef_path term video from source-address 11.19.142.0/24
set firewall family inet filter ef_path term video from source-address 11.19.144.0/24
set firewall family inet filter ef_path term video then topology video
set firewall family inet filter ef_path term default then accept
set class-of-service interfaces fe-1/2/0 unit 22 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/1 unit 25 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/2 unit 34 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/3 unit 37 classifiers inet-precedence default
```

Device P3

```
set interfaces fe-1/2/0 unit 10 description to-PE1
set interfaces fe-1/2/0 unit 10 family inet filter input ef_path
set interfaces fe-1/2/0 unit 10 family inet address 10.0.0.10/30
set interfaces fe-1/2/1 unit 14 description to-P1
set interfaces fe-1/2/1 unit 14 family inet filter input ef_path
set interfaces fe-1/2/1 unit 14 family inet address 10.0.0.14/30
set interfaces fe-1/2/2 unit 21 description to-P2
set interfaces fe-1/2/2 unit 21 family inet filter input ef_path
set interfaces fe-1/2/2 unit 21 family inet address 10.0.0.21/30
set interfaces fe-1/2/3 unit 29 description to-P4
set interfaces fe-1/2/3 unit 29 family inet filter input ef_path
set interfaces fe-1/2/3 unit 29 family inet address 10.0.0.29/30
set interfaces lo0 unit 111 family inet address 10.255.165.111/32 primary
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.111
set protocols bgp group ibgp family inet unicast topology voice community target:40:40
set protocols bgp group ibgp family inet unicast topology video community target:50:50
set protocols bgp group ibgp neighbor 10.255.165.93
set protocols bgp group ibgp neighbor 10.255.165.113
set protocols bgp group ibgp neighbor 10.255.165.203
set protocols bgp group ibgp neighbor 10.255.165.95
set protocols bgp group ibgp neighbor 10.255.165.99
set protocols ospf topology voice topology-id 126
set protocols ospf topology video topology-id 52
set protocols ospf area 0.0.0.0 interface fe-1/2/3.29 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/3.29 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/3.29 topology video
set protocols ospf area 0.0.0.0 interface fe-1/2/2.21 metric 10
```

```
set protocols ospf area 0.0.0.0 interface fe-1/2/2.21 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/2.21 topology video metric 20
set protocols ospf area 0.0.0.0 interface fe-1/2/1.14 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/1.14 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/1.14 topology video disable
set protocols ospf area 0.0.0.0 interface fe-1/2/0.10 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/0.10 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/0.10 topology video
set protocols ospf area 0.0.0.0 interface lo0.111 passive
set routing-options autonomous-system 100
set routing-options topologies family inet topology voice
set routing-options topologies family inet topology video
set firewall family inet filter ef_path term ef from forwarding-class expedited-forwarding
set firewall family inet filter ef_path term ef then topology voice
set firewall family inet filter ef_path term video from source-address 11.19.132.0/24
set firewall family inet filter ef_path term video from source-address 11.19.133.0/24
set firewall family inet filter ef_path term video from source-address 11.19.142.0/24
set firewall family inet filter ef_path term video from source-address 11.19.144.0/24
set firewall family inet filter ef_path term video then topology video
set firewall family inet filter ef_path term default then accept
set class-of-service interfaces fe-1/2/0 unit 10 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/1 unit 14 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/2 unit 21 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/3 unit 29 classifiers inet-precedence default
```

Device P4

```
set interfaces fe-1/2/0 unit 18 description to-P1
set interfaces fe-1/2/0 unit 18 family inet filter input ef_path
set interfaces fe-1/2/0 unit 18 family inet address 10.0.0.18/30
set interfaces fe-1/2/1 unit 26 description to-P2
set interfaces fe-1/2/1 unit 26 family inet filter input ef_path
set interfaces fe-1/2/1 unit 26 family inet address 10.0.0.26/30
set interfaces fe-1/2/2 unit 30 description to-P3
set interfaces fe-1/2/2 unit 30 family inet filter input ef_path
set interfaces fe-1/2/2 unit 30 family inet address 10.0.0.30/30
set interfaces fe-1/2/3 unit 41 description to-PE2
set interfaces fe-1/2/3 unit 41 family inet filter input ef_path
set interfaces fe-1/2/3 unit 41 family inet address 10.0.0.41/30
set interfaces lo0 unit 95 family inet address 10.255.165.95/32 primary
```

```
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.95
set protocols bgp group ibgp family inet unicast topology voice community target:40:40
set protocols bgp group ibgp family inet unicast topology video community target:50:50
set protocols bgp group ibgp neighbor 10.255.165.93
set protocols bgp group ibgp neighbor 10.255.165.113
set protocols bgp group ibgp neighbor 10.255.165.203
set protocols bgp group ibgp neighbor 10.255.165.111
set protocols bgp group ibgp neighbor 10.255.165.99
set protocols ospf topology voice topology-id 126
set protocols ospf topology video topology-id 52
set protocols ospf area 0.0.0.0 interface fe-1/2/2.30 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/2.30 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/2.30 topology video
set protocols ospf area 0.0.0.0 interface fe-1/2/0.18 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/0.18 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/0.18 topology video disable
set protocols ospf area 0.0.0.0 interface fe-1/2/0.18 topology video metric 20
set protocols ospf area 0.0.0.0 interface fe-1/2/1.26 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/1.26 topology voice disable
set protocols ospf area 0.0.0.0 interface fe-1/2/1.26 topology video disable
set protocols ospf area 0.0.0.0 interface fe-1/2/3.41 metric 10
set protocols ospf area 0.0.0.0 interface fe-1/2/3.41 topology voice
set protocols ospf area 0.0.0.0 interface fe-1/2/3.41 topology video
set protocols ospf area 0.0.0.0 interface lo0.95 passive
set routing-options autonomous-system 100
set routing-options topologies family inet topology voice
set routing-options topologies family inet topology video
set firewall family inet filter ef_path term ef from forwarding-class expedited-forwarding
set firewall family inet filter ef_path term ef then topology voice
set firewall family inet filter ef_path term video from source-address 11.19.132.0/24
set firewall family inet filter ef_path term video from source-address 11.19.133.0/24
set firewall family inet filter ef_path term video from source-address 11.19.142.0/24
set firewall family inet filter ef_path term video from source-address 11.19.144.0/24
set firewall family inet filter ef_path term video then topology video
set firewall family inet filter ef_path term default then accept
set class-of-service interfaces fe-1/2/0 unit 18 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/1 unit 26 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/2 unit 30 classifiers inet-precedence default
set class-of-service interfaces fe-1/2/3 unit 41 classifiers inet-precedence default
```

Configuring Device CE1

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 CE1:

1. Configure the interfaces.

   For demonstration purposes, the example places an Ethernet interface into loopback mode and configures several addresses on this loopback interface. The addresses are then announced to the network.

   ```
   [edit interfaces]
   user@CE1# set fe-0/1/0 fastether-options loopback
   user@CE1# set fe-0/1/0 unit 0 family inet address 11.19.130.1/24
   user@CE1# set fe-0/1/0 unit 0 family inet address 11.19.131.1/24
   user@CE1# set fe-0/1/0 unit 0 family inet address 11.19.132.1/24
   user@CE1# set fe-1/2/0 unit 1 description to-PE1
   user@CE1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30
   user@CE1# set lo0 unit 97 family inet address 10.255.165.97/32 primary
   ```

2. Configure the external BGP (EBGP) connection to Device PE1.

   ```
   [edit protocols bgp group ebgp]
   user@CE1# set type external
   user@CE1# set local-address 10.0.0.1
   user@CE1# set peer-as 100
   user@CE1# set neighbor 10.0.0.2
   ```

3. Configure the routing policy that announces the addresses that are configured on interface fe-0/1/0.

   ```
   [edit policy-options policy-statement inject_directs]
   user@CE1# set term a from protocol direct
   user@CE1# set term a from interface fe-0/1/0.0
   user@CE1# set term a then next policy
   user@CE1# set term a then accept
   user@CE1# set term b then reject
   ```

4. Configure the routing policy that tags voice routes with the video community attribute, and video routes with the voice community attribute.

```
[edit policy-options policy-statement set_community]
user@CE1# set term a from route-filter 11.19.130.0/24 exact
user@CE1# set term a from route-filter 11.19.131.0/24 exact
user@CE1# set term a then community add voice
user@CE1# set term a then accept
user@CE1# set term b from route-filter 11.19.132.0/24 exact
user@CE1# set term b from route-filter 11.19.133.0/24 exact
user@CE1# set term b then community add video
user@CE1# set term b then accept
user@CE1# set term default then accept
[edit policy-options community]
user@CE1# set video members target:50:50
user@CE1# set voice members target:40:40
```

5. Apply the **set_community** export policy so that direct routes are exported from the routing table into BGP.

   Apply the **inject_directs** export policy to announce the addresses that are configured on interface fe-0/1/0.

```
[edit protocols bgp group ebgp]
user@CE1# set export set_community
user@CE1# set export inject_directs
```

6. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@CE1# set autonomous-system 101
```

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@CE1# show interfaces
fe-0/1/0 {
    fastether-options {
        loopback;
    }
    unit 0 {
        family inet {
```

```
            address 11.19.130.1/24;
            address 11.19.131.1/24;
            address 11.19.132.1/24;
        }
    }
}
fe-1/2/0 {
    unit 1 {
        description to-PE1;
        family inet {
            address 10.0.0.1/30;
        }
    }
}
lo0 {
    unit 97 {
        family inet {
            address 10.255.165.97/32 {
                primary;
            }
        }
    }
}
```

```
user@CE1# show protocols
bgp {
    group ebgp {
        type external;
        local-address 10.0.0.1;
        export [ set_community inject_directs ];
        peer-as 100;
        neighbor 10.0.0.2;
    }
}
```

```
user@CE1# show policy-options
policy-statement inject_directs {
    term a {
        from {
            protocol direct;
            interface fe-0/1/0.0;
        }
        then {
```

```
            next policy;
            accept;
        }
    }
    term b {
        then reject;
    }
}
policy-statement set_community {
    term a {
        from {
            route-filter 11.19.130.0/24 exact;
            route-filter 11.19.131.0/24 exact;
        }
        then {
            community add voice;
            accept;
        }
    }
    term b {
        from {
            route-filter 11.19.132.0/24 exact;
            route-filter 11.19.133.0/24 exact;
        }
        then {
            community add video;
            accept;
        }
    }
    term default {
        then accept;
    }
}
community video members target:50:50;
community voice members target:40:40;
```

```
user@CE1# show routing-options
autonomous-system 101;
```

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

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.

   The forwarding plane uses a firewall filter to indicate which topology forwarding table traffic should use. In this case, you must configure a firewall filter on all interfaces related to routing topologies. In general, all multitopology OSPF interfaces in the core where topologies are configured have input firewall filters. In addition, the ingress interfaces, where traffic from a CE device enters a PE device toward the core, have firewall filters configured. This configuration on Device PE1 shows a firewall filter applied to the ingress interface (connected to the CE device) and to the two core-facing interfaces (connected to Device P1 and Device P3).

   ```
   [edit interfaces]
   user@PE1# set fe-1/2/0 unit 2 description to-CE1
   user@PE1# set fe-1/2/0 unit 2 family inet filter input ef_path
   user@PE1# set fe-1/2/0 unit 2 family inet address 10.0.0.2/30
   user@PE1# set fe-1/2/1 unit 6 description to-P1
   user@PE1# set fe-1/2/1 unit 6 family inet filter input ef_path
   user@PE1# set fe-1/2/1 unit 6 family inet address 10.0.0.6/30
   user@PE1# set fe-1/2/2 unit 9 description to-P3
   user@PE1# set fe-1/2/2 unit 9 family inet filter input ef_path
   user@PE1# set fe-1/2/2 unit 9 family inet address 10.0.0.9/30
   user@PE1# set lo0 unit 93 family inet address 10.255.165.93/32 primary
   ```

2. Configure the autonomous system (AS) number.

   ```
   [edit routing-options]
   user@PE1# set autonomous-system 100
   ```

3. Configure BGP.

   ```
   [edit protocols bgp group ibgp]
   user@PE1# set type internal
   user@PE1# set local-address 10.255.165.93
   user@PE1# set neighbor 10.255.165.111
   user@PE1# set neighbor 10.255.165.203
   user@PE1# set neighbor 10.255.165.113
   user@PE1# set neighbor 10.255.165.95
   user@PE1# set neighbor 10.255.165.99
   [edit protocols bgp group ebgp]
   ```

```
user@PE1# set type external
user@PE1# set local-address 10.0.0.2
user@PE1# set peer-as 101
user@PE1# set neighbor 10.0.0.1
```

4. Configure a next-hop self routing policy to ensure that the IBGP devices use the loopback address on Device PE1 as the next-hop address on all IBGP route advertisements.

   This way, Device PE1 serves as the gateway router for EBGP routes.

   ```
   [edit policy-options policy-statement nhs]
   user@PE1# set then next-hop self
   ```

5. Apply the next-hop self policy to the IBGP sessions.

   ```
   [edit protocols bgp group ibgp]
   user@PE1# set export nhs
   ```

6. Configure the voice and video topologies, which enable you to use these topologies with OSPF and BGP.

   The names **voice** and **video** are local to the router. The names are not propagated beyond this router. However, for management purposes, a consistent naming scheme across routers in a multitopology environment is convenient.

   ```
   [edit routing-options topologies family inet]
   user@PE1# set topology voice
   user@PE1# set topology video
   ```

7. Apply the community tags to identify the voice and video topologies by configuring a routing topology name and BGP community value.

   In Junos OS, multitopology support for BGP is based on the community value in a BGP route. This configuration determines the association between a topology and one or more community values and populates the topology routing tables. Arriving BGP updates that have a matching community value are replicated in the associated topology routing table. You decide which BGP community values are associated with a given topology.

   This configuration causes BGP updates that are received with community value **target:40:40** to be added into topology routing table **:voice.inet.0** (in addition to the default routing table **inet.0**). Updates that are received with community value **target:50:50** are added into topology routing table **:video.inet.0** (in addition to the default routing table **inet.0**).

```
[edit protocols bgp group ibgp family inet unicast]
user@PE1# set topology voice community target:40:40
user@PE1# set topology video community target:50:50
[edit protocols bgp group ebgp family inet unicast]
user@PE1# set topology voice community target:40:40
user@PE1# set topology video community target:50:50
```

8. Enable and disable multitopology OSPF on particular interfaces.

   Enable multitopology OSPF designations only on desired interfaces, as shown in Figure 5 on page 21. On interface fe-1/2/1.6 facing Device P1, enable the voice topology, and disable the video topology. On interface fe-1/2/2.9 facing Device P3, enable the video topology, and disable the voice topology.

   When a topology ID is configured under OSPF, the topology is automatically enabled on all interfaces under OSPF. To disable a topology or to add a metric, you must add an explicit configuration.

   For readability purposes, each topology is configured under each desired OSPF interface even though this default behavior occurs when the topology ID is configured. Configure higher metric values on a link to make the link less preferred than another available link.

```
[edit protocols ospf ]
user@PE1# set topology voice topology-id 126
user@PE1# set topology video topology-id 52
[edit protocols ospf area 0.0.0.0]
user@PE1# set interface fe-1/2/1.6 metric 10
user@PE1# set interface fe-1/2/1.6 topology video disable
user@PE1# set interface fe-1/2/1.6 topology voice
user@PE1# set interface fe-1/2/2.9 metric 10
user@PE1# set interface fe-1/2/2.9 topology voice disable
user@PE1# set interface fe-1/2/2.9 topology video
user@PE1# set interface lo0.93 passive
```

9. Configure the firewall filter.

   After routing topologies are configured, traffic must go through a firewall filter to make use of routing topology forwarding tables. For basic routing topologies, where traffic is first entering the core network, apply an input firewall filter to the ingress interface. Additionally, add firewall filters to interfaces where multitopology OSPF is configured. All routers must use the same firewall filter to associate packets with a topology to ensure consistent forwarding and to avoid routing loops or packet loss.

   The forwarding plane handles traffic as it enters the router and exits out a particular interface. To inspect traffic and use a specified topology forwarding table to perform next-hop lookups, configure an input firewall filter on each interface where routing topology support is desired. Use a regular firewall filter to identify packet characteristics.

In general, for application-level differentiation, it is convenient to use DiffServ code points (DSCPs). When there is a firewall filter match, the firewall instructs the route lookup to use a particular topology forwarding table. Packet attributes are identified in the **from** clause, followed by a **then** clause indicating the topology forwarding table for use in forwarding next-hop lookups. This configuration notifies the router which traffic uses a routing topology forwarding table and which traffic uses the default forwarding table. The last term, which is named **default**, specifies the use of the default forwarding table.

These firewall configurations show source addresses and DSCPs used to sort voice, video, and default traffic. DSCPs are practical because you can set them at or near a CE device and because the information is intact across the network. For instance, here class of service (CoS) is configured for expedited traffic. DSCPs are also practical when the same IP address is used for different applications.

```
[edit firewall family inet filter ef_path]
user@PE1# set term ef from forwarding-class expedited-forwarding
user@PE1# set term ef then topology voice
user@PE1# set term video from source-address 11.19.132.0/24
user@PE1# set term video from source-address 11.19.133.0/24
user@PE1# set term video from source-address 11.19.142.0/24
user@PE1# set term video from source-address 11.19.144.0/24
user@PE1# set term video then topology video
user@PE1# set term default then accept
```

10. Enable CoS on the interfaces.

```
[edit class-of-service interfaces]
user@PE1# set fe-1/2/0 unit 2 classifiers inet-precedence default
user@PE1# set fe-1/2/1 unit 6 classifiers inet-precedence default
user@PE1# set fe-1/2/2 unit 9 classifiers inet-precedence default
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, **show routing-options**, **show firewall**, and **show class-of-service** 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 2 {
        description to-CE1;
        family inet {
            filter {
```

```
            input ef_path;
          }
          address 10.0.0.2/30;
        }
    }
}
fe-1/2/1 {
    unit 6 {
        description to-P1;
        family inet {
            filter {
                input ef_path;
            }
            address 10.0.0.6/30;
        }
    }
}
fe-1/2/2 {
    unit 9 {
        description to-P3;
        family inet {
            filter {
                input ef_path;
            }
            address 10.0.0.9/30;
        }
    }
}
lo0 {
    unit 93 {
        family inet {
            address 10.255.165.93/32 {
                primary;
            }
        }
    }
}
```

```
user@PE1# show protocols
bgp {
    group ibgp {
        type internal;
        local-address 10.255.165.93;
        family inet {
```

```
            unicast {
                topology voice {
                    community target:40:40;
                }
                topology video {
                    community target:50:50;
                }
            }
        }
        export nhs;
        neighbor 10.255.165.111;
        neighbor 10.255.165.203;
        neighbor 10.255.165.113;
        neighbor 10.255.165.95;
        neighbor 10.255.165.99;
    }
    group ebgp {
        type external;
        local-address 10.0.0.2;
        family inet {
            unicast {
                topology voice {
                    community target:40:40;
                }
                topology video {
                    community target:50:50;
                }
            }
        }
        peer-as 101;
        neighbor 10.0.0.1;
    }
}
ospf {
    topology voice topology-id 126;
    topology video topology-id 52;
    area 0.0.0.0 {
        interface fe-1/2/1.6 {
            metric 10;
            topology video disable;
            topology voice;
        }
        interface fe-1/2/2.9 {
            metric 10;
```

```
            topology voice disable;
            topology video;
        }
        interface lo0.93 {
            passive;
        }
    }
}
```

```
user@PE1# show policy-options
policy-statement nhs {
    then {
        next-hop self;
    }
}
```

```
user@PE1# show routing-options
autonomous-system 100;
topologies {
    family inet {
        topology voice;
        topology video;
    }
}
```

```
user@PE1# show firewall
family inet {
    filter ef_path {
        term ef {
            from {
                forwarding-class expedited-forwarding;
            }
            then topology voice;
        }
        term video {
            from {
                source-address {
                    11.19.132.0/24;
                    11.19.133.0/24;
                    11.19.142.0/24;
                    11.19.144.0/24;
                }
```

```
        }
        then topology video;
      }
      term default {
        then accept;
      }
    }
}
```

```
user@PE1# show class-of-service
interfaces {
   fe-1/2/0 {
      unit 2 {
         classifiers {
            inet-precedence default;
         }
      }
      unit 6 {
         classifiers {
            inet-precedence default;
         }
      }
      unit 9 {
         classifiers {
            inet-precedence default;
         }
      }
   }
}
```

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

Verification

IN THIS SECTION

- Verifying the OSPF Interfaces | **44**
- Verifying the Routes | **45**
- Checking the Resolving BGP Next Hops | **46**

- Examining the Protocol Next Hop | 48
- Verifying the OSPF Neighbor | 49
- Checking the Router LSA | 49
- Checking How Traffic Traverses the Network | 50

Confirm that the configuration is working properly.

Verifying the OSPF Interfaces

Purpose

Verify that the OSPF interfaces are configured to belong to one or more topologies.

Action

From operational mode, enter the **show (ospf | ospf3) interface *interface-name* detail** command.

```
user@PE1> show ospf interface fe-1/2/1.6 detail
```

```
Interface           State   Area            DR ID           BDR ID          Nbrs
fe-1/2/1.6          DR      0.0.0.0         10.255.165.93   10.255.165.99      1
  Type: LAN, Address: 10.0.0.6, Mask: 255.255.255.252, MTU: 1500, Cost: 10
  DR addr: 10.0.0.6, BDR addr: 10.0.0.5, Priority: 128
  Adj count: 1
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
  Protection type: None
  Topology default (ID 0) -> Cost: 10
  Topology video (ID 52) -> Disabled, Cost: 10
  Topology voice (ID 126) -> Cost: 10
```

Meaning

This output shows that the voice topology was added to the fe-1/2/1.6 interface on Device PE1. The topology name is voice, and the MT-ID is 126. The video topology is disabled on this interface. The cost of the interface is 10.

The Router-LSA originated and flooded by the router includes all relevant topology information for specific interfaces, such as MT-ID and metric. If MTR is not configured on an OSPF interface, then the Router-LSA does not include any topology information for that interface. OSPF neighbors might or might not support multitopology OSPF. That is, a particular link is not used to calculate OSPF routes for a topology unless

routers at both ends of the link announce that link as part of the topology. If multitopology OSPF is not supported in neighboring OSPF routers or is not configured to do so, then topology information in LSAs received by the neighbor is ignored.

Verifying the Routes

Purpose

Make sure that the routes are in the expected routing tables and that the expected communities are attached to the routes.

Action

From operational mode, enter the **show route detail** command on Device PE1.

`user@PE1>` **show route 11.19.130.0/24 detail**

```
inet.0: 29 destinations, 30 routes (29 active, 0 holddown, 0 hidden)
11.19.130.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Router, Next hop index: 812
                Address: 0xb9f064c
                Next-hop reference count: 22
                Source: 10.0.0.1
                Next hop: 10.0.0.1 via fe-1/2/0.2, selected
                Session Id: 0x600004
                State: <Active Ext>
                Local AS:   100 Peer AS:   101
                Age: 3d 21:44:07
                Task: BGP_101.10.0.0.1+51873
                Announcement bits (3): 0-KRT 3-BGP_RT_Background 4-Resolve tree 3

                AS path: 101 I
                Communities: target:40:40
                Accepted
                Localpref: 100
                Router ID: 10.255.165.97
                Secondary Tables: :voice.inet.0

:voice.inet.0: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)

11.19.130.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Router, Next hop index: 812
                Address: 0xb9f064c
```

```
                Next-hop reference count: 22
                Source: 10.0.0.1
                Next hop: 10.0.0.1 via fe-1/2/0.2, selected
                Session Id: 0x600004
                State: <Secondary Active IndepResolution Ext>
                Local AS:   100 Peer AS:   101
                Age: 3d 21:44:07
                Task: BGP_101.10.0.0.1+51873
                Announcement bits (2): 0-KRT 1-Resolve tree 1
                AS path: 101 I
                Communities: target:40:40
                Accepted
                Localpref: 100
                Router ID: 10.255.165.97
                Primary Routing Table inet.0
```

Meaning

This output shows BGP route 11.19.130.0/24 with community value target:40:40. Because the route matches the criteria for the voice topology, it is added to both the default and voice topology routing tables (**inet.0** and **:voice.inet.0**). Device PE1 learns the route from Device CE1 through EBGP and then injects the route into IBGP.

Checking the Resolving BGP Next Hops

Purpose

Check the protocol next hop and forwarding next hop.

Action

From operational mode, enter the **show route detail** command on Device PE2.

```
user@PE2> show route 11.19.130.0/24 detail
```

```
inet.0: 29 destinations, 30 routes (29 active, 0 holddown, 0 hidden)
11.19.130.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0xb9f0e04
                Next-hop reference count: 12
                Source: 10.255.165.93
                Next hop type: Router, Next hop index: 262153
                Next hop: 10.0.0.37 via fe-1/2/0.38
                Session Id: 0x700004
```

```
                Next hop: 10.0.0.41 via fe-1/2/1.42, selected
                Session Id: 0x700005
                Protocol next hop: 10.255.165.93
                Indirect next hop: bb8c000 262154 INH Session ID: 0x700007
                State: <Active Int Ext>
                Local AS:   100 Peer AS:   100
                Age: 3d 4:27:40         Metric2: 30
                Task: BGP_100.10.255.165.93+179
                Announcement bits (3): 0-KRT 3-BGP_RT_Background 4-Resolve tree 3

                AS path: 101 I
                Communities: target:40:40
                Accepted
                Localpref: 100
                Router ID: 10.255.165.93
                Secondary Tables: :voice.inet.0

:voice.inet.0: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)

11.19.130.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0xb9f0f34
                Next-hop reference count: 6
                Source: 10.255.165.93
                Next hop type: Router, Next hop index: 1188
                Next hop: 10.0.0.37 via fe-1/2/0.38, selected
                Session Id: 0x700004
                Protocol next hop: 10.255.165.93
                Indirect next hop: bb8c1d8 262177 INH Session ID: 0x700007
                State: <Secondary Active IndepResolution Int Ext>
                Local AS:   100 Peer AS:   100
                Age: 3d 2:00:20         Metric2: 30
                Task: BGP_100.10.255.165.93+179
                Announcement bits (2): 0-KRT 1-Resolve tree 1
                AS path: 101 I
                Communities: target:40:40
                Accepted
                Localpref: 100
                Router ID: 10.255.165.93
                Primary Routing Table inet.0
```

Meaning

A typical IBGP core has BGP routes with protocol next hops that resolve using the underlying IGP routes. IBGP routes in a topology routing table have protocol next-hop IP addresses. By default, the same topology routing table is used to look up and resolve the protocol next-hop IP address to a forwarding next hop. This output from Device PE2 shows the same BGP route as seen in the previous example: 11.19.130.0/24. The route is being shown from a different perspective, that is, from Device PE2 as an IBGP route. Similarly, this IBGP route is added to both **inet.0** and **:voice.inet.0** on Device PE2. However, while each route has the same protocol next hop, each route has a different forwarding next hop (ge-0/0/3.0 instead of ge-0/1/4.0). The reason for this difference is when the protocol next-hop IP address 10.255.165.93 is resolved, it uses the corresponding routing table (**inet.0** or **:voice.inet.0**) to look up the protocol next hop.

Examining the Protocol Next Hop

Purpose

Check the protocol next hop and forwarding next hop.

Action

From operational mode, enter the **show route** command on Device PE2.

`user@PE2>` **show route 10.255.165.93**

```
inet.0: 29 destinations, 30 routes (29 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.165.93/32   *[OSPF/10] 3d 04:37:26, metric 30
                    > to 10.0.0.37 via fe-1/2/0.38
                      to 10.0.0.41 via fe-1/2/1.42

:voice.inet.0: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.165.93/32   *[OSPF/10] 3d 02:10:04, metric 30
                    > to 10.0.0.37 via fe-1/2/0.38

:video.inet.0: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.165.93/32   *[OSPF/10] 3d 02:03:16, metric 30
                    > to 10.0.0.41 via fe-1/2/1.42
```

Meaning

This output from Device PE2 shows the protocol next hop of 11.19.130.0/24, which is IP address 10.255.165.93, thus further demonstrating how IBGP route 11.19.130.0/24 resolves its protocol next hop. The forwarding next hops of 10.255.165.93 match the IBGP forwarding next hops of route

11.19.130/24 as shown in the previous example. Observe here that the IP address 10.255.165.93 is also in routing table **:video.inet.0**. This address is the loopback address of Device PE1, and as such, resides in all three routing tables. This example also shows how traffic entering Device PE2 destined to 11.19.130.0/24 exits out different interfaces depending on its associated topology. The actual traffic is marked in such a way that a firewall filter can direct the traffic to use a particular topology routing table.

Verifying the OSPF Neighbor

Purpose

Make sure that the expected topologies are enabled on the OSPF neighbor.

Action

From operational mode, enter the **show ospf neighbor 10.0.0.21 extensive** command on Device P2.

```
user@P2> show ospf neighbor 10.0.0.21 extensive
```

```
Address          Interface              State     ID               Pri  Dead
10.0.0.21        fe-1/2/0.22            Full      10.255.165.111   128    39
  Area 0.0.0.0, opt 0x52, DR 10.0.0.22, BDR 10.0.0.21
  Up 3d 06:09:50, adjacent 3d 06:09:50
   Topology default (ID 0) -> Bidirectional
   Topology video (ID 52) -> Bidirectional
```

Meaning

This Device P2 output shows OSPF neighbor PE2 (10.0.0.21), where multitopology OSPF default and video are multitopology OSPF participants. The **Bidirectional** flag shows that the neighbor is configured using the same multitopology OSPF ID.

Checking the Router LSA

Purpose

Check the links where video and voice topologies are enabled.

Action

From operational mode, enter the **show ospf database lsa-id 10.255.165.203 extensive** command on Device P2.

```
user@P2> show ospf database lsa-id 10.255.165.203 extensive
```

```

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
```

```
Router   10.255.165.203   10.255.165.203   0x80000063  1552  0x22 0xdff3  80
  bits 0x0, link count 3
  id 10.255.165.203, data 255.255.255.255, Type Stub (3)
    Topology count: 2, Default metric: 0
    Topology video (ID 52) -> Metric: 0
    Topology voice (ID 126) -> Metric: 0
  id 10.0.0.38, data 10.0.0.38, Type Transit (2)
    Topology count: 2, Default metric: 10
    Topology video (ID 52) -> Metric: 200
    Topology voice (ID 126) -> Metric: 10
  id 10.0.0.42, data 10.0.0.42, Type Transit (2)
    Topology count: 1, Default metric: 10
    Topology video (ID 52) -> Metric: 10
  Topology default (ID 0)
    Type: Transit, Node ID: 10.0.0.42
      Metric: 10, Bidirectional
    Type: Transit, Node ID: 10.0.0.38
      Metric: 10, Bidirectional
  Topology video (ID 52)
    Type: Transit, Node ID: 10.0.0.42
      Metric: 10, Bidirectional
    Type: Transit, Node ID: 10.0.0.38
      Metric: 200, Bidirectional
  Topology voice (ID 126)
    Type: Transit, Node ID: 10.0.0.38
      Metric: 10, Bidirectional
  Aging timer 00:34:08
  Installed 00:25:49 ago, expires in 00:34:08, sent 00:25:47 ago
  Last changed 3d 01:45:51 ago, Change count: 10
```

Meaning

This Device P2 output shows the Router-LSA originated by Device PE2. The LSA shows links where video and voice topologies are enabled (in addition to the default topology).

Checking How Traffic Traverses the Network

Purpose

Make sure that the expected paths are used.

Action

From operational mode, enter the **traceroute** command on Device CE1.

The first example output shows a traceroute over the voice topology goes from Device CE1 to Device CE2 where DSCPs are set. The routes are resolved over **:voice.inet.0**. This traceroute path follows the voice path CE1-PE1-P1-P2-PE2-CE2

```
user@CE1> traceroute 11.19.140.1 source 11.19.130.1 tos 160
```

```
traceroute to 11.19.140.1 (11.19.140.1) from 11.19.130.1, 30 hops max, 40 byte
packets
 1  10.0.0.2 (10.0.0.2)  2.015 ms  1.924 ms  1.770 ms
 2  10.0.0.5 (10.0.0.5)  1.890 ms  1.010 ms  0.974 ms
 3  10.0.0.34 (10.0.0.34)  0.986 ms  1.031 ms  0.973 ms
 4  10.0.0.38 (10.0.0.38)  1.213 ms  1.065 ms  1.154 ms
 5  11.19.140.1 (11.19.140.1)  1.696 ms  4.286 ms  1.332 ms
```

This output shows a traceroute from Device CE1 to Device CE2 for voice where no DSCPs are set. The routes are resolved over **inet.0**, and the resulting path is different from the previous case where the DSCPs are set. This traceroute path follows the default path CE1-PE1-P4-PE2-CE2.

```
user@CE1> traceroute 11.19.140.1 source 11.19.130.1
```

```
traceroute to 11.19.140.1 (11.19.140.1) from 11.19.130.1, 30 hops max, 40 byte
packets
 1  10.0.0.2 (10.0.0.2)  1.654 ms  1.710 ms  1.703 ms
 2  10.0.0.5 (10.0.0.5)  1.790 ms  1.045 ms  0.975 ms
 3  10.0.0.18 (10.0.0.18)  0.989 ms  1.041 ms  0.983 ms
 4  10.0.0.42 (10.0.0.42)  0.994 ms  1.036 ms  1.002 ms
 5  11.19.140.1 (11.19.140.1)  1.329 ms  2.248 ms  2.225 ms
```

This output shows a traceroute from Device CE1 to Device CE2 for video traffic where the firewall filter is based on the destination address. The routes are resolved over **:video.inet.0** . This traceroute follows the video path CE1-PE1-P3-P4-PE2-CE2.

```
user@CE1> traceroute 11.19.142.1 source 11.19.132.1
```

```
traceroute to 11.19.142.1 (11.19.142.1) from 11.19.132.1, 30 hops max, 40 byte
packets
```

```
 1  10.0.0.2 (10.0.0.2)  1.126 ms  1.300 ms  0.995 ms
 2  10.0.0.10 (10.0.0.10)  0.981 ms  1.018 ms  0.991 ms
 3  10.0.0.30 (10.0.0.30)  0.997 ms  1.886 ms  1.952 ms
 4  10.0.0.42 (10.0.0.42)  1.800 ms  1.038 ms  0.980 ms
 5  11.19.142.1 (11.19.142.1)  1.367 ms  1.352 ms  1.328 ms
```

This output shows a traceroute from Device CE1 to Device CE2 for video where DSCPs are set. The DSCP bits are directing Device PE1 to use the topology table **:voice.inet.0**. Because there is no entry in the voice routing table for video routes, traffic is dropped.

`user@CE1>` **traceroute 11.19.142.1 source 11.19.132.1 tos 160**

```
traceroute to 11.19.142.1 (11.19.142.1) from 11.19.132.1, 30 hops max, 40 byte
packets
 1  10.0.0.2 (10.0.0.2)  1.135 ms !N  1.007 ms !N  0.954 ms !N
```

RELATED DOCUMENTATION

Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 18

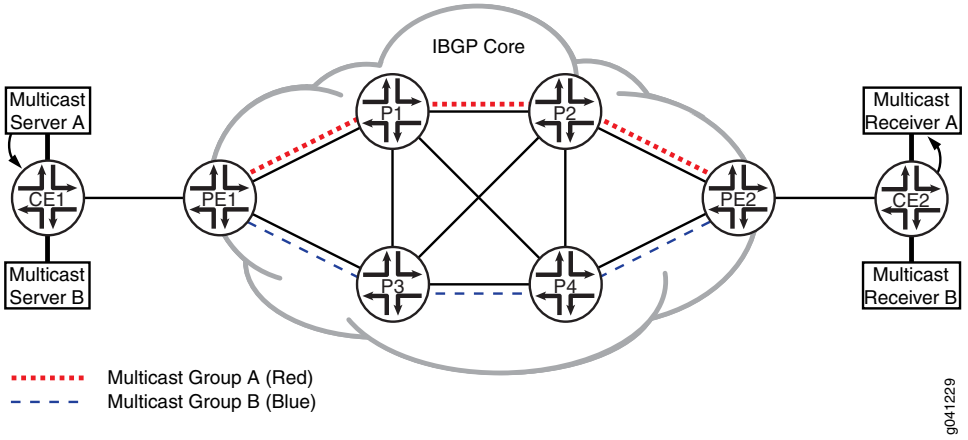
Understanding Multitopology Routing in Conjunction with PIM

Protocol Independent Multicast (PIM), in conjunction with multitopology routing extensions to OSPF (multitopology OSPF) and BGP, can direct multicast traffic over particular paths based on traffic characteristics.

Junos OS provides a mechanism whereby multicast traffic traverses user-specified topology paths based on the sender's source address. Multitopology routing (MTR) is used for OSPF, BGP, and route resolution over the specified topology routing tables. OSPF and BGP independently populate the routing table used by PIM. Firewall filters are not required because the multicast forwarding plane uses the multicast tree after it has been built.

Figure 6 on page 53 shows a diagram of routing topology paths, where the dashed lines are associated with multicast group A (topology red), and the dotted lines are associated with multicastgroup B (topology blue).

Figure 6: Core Links Configured to Prefer Specified Routing Topologies



Two copies of the same stream enter Device PE1 and then traverse separate paths over the internal BGP (IBGP) core.

This solution leverages Junos OS features that allow particular routing tables to perform route resolution using specified routing tables.

The configuration includes a combination of the following features:

- BGP communities
- Separate IBGP next hops belonging to user-specified OSPF routing topologies
- Route resolution over user-specified topology routing tables
- A separate routing table (**inet.2**) for multicast protocols

Commonly, networks use a separate routing table for multicast. In Junos OS, the multicast routing table is **inet.2**. Routing topologies are grouped based on BGP communities. Each group represents a set of IP addresses associated with multicast servers and receivers. Primarily, the group must be related to the set of servers because the multicast receivers initiate tree creation toward these servers. Multicast traffic directed downstream toward receivers uses the previously created PIM tree, and therefore the forwarding plane does not need to know about routing topologies.

PIM uses the **inet.2** routing table for lookups of multicast source addresses. These IP addresses used for tree creation are IP unicast addresses. The customer edge (CE) routers, nearest to the multicast servers, announce the multicast source IP addresses to the provider edge (PE) routers using external BGP (EBGP). They are announced with both **family inet** unicast and **family inet multicast**, thus causing the BGP route to be added to the default routing table **inet.0** and to **inet.2**.

Both versions of the route are injected by the PE router into IBGP. Each BGP route injected into IBGP has a specific protocol next hop. Junos OS provides the flexibility to set the protocol next hop when exporting the route into IBGP. For instance, a next-hop self can be set with an export policy configuration. You can also set the protocol next hop to a route associated with a specified topology routing table.

Keeping in mind that an EBGP route can have a community associated with a routing topology, you can conveniently configure a policy to use this community to designate which protocol next hop should be set when exporting the IBGP route into **inet.2**. As such, a specific protocol next-hop IP address is required for each topology on each router injecting IBGP routes. You can configure multiple secondary loopback IP addresses on a router to be used as protocol next-hop addresses.

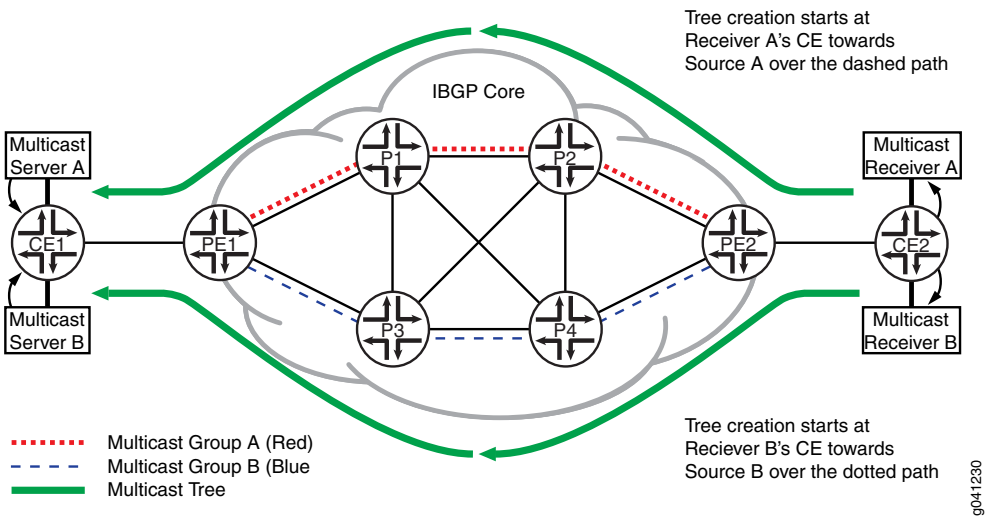
A group of BGP routes associated with a routing topology use the same unique protocol next hop. For instance, if you configure a PE router to handle two routing topologies, you would also configure two unique nonprimary addresses under loopback interface lo0. Next, associate each nonprimary loopback IP address with a topology for inclusion in the associated topology routing table. Configure the loopback IP address and topology under an OSPF interface statement. You must specifically disable all other topologies known to OSPF for two reasons. First, the loopback address specific to a topology must reside in only one topology routing table. Second, once the topology is added to OSPF, the topology defaults to being enabled on all subsequent interfaces under OSPF.

You can specify up to two routing tables in the resolution configuration. A key element to this solution is that the protocol next-hop address resides in only one topology table. That is, the protocol next hop belongs to a remote PE secondary loopback address and is injected into only one topology table. The route resolution scheme first checks the topology table for the protocol next-hop address. If the address is found, it uses this entry. If it is not found, the resolution scheme then checks the second topology table. Hence, only one topology table is used for each protocol nexthop address.

Links can support all routing topologies to provide a backup path should a primary multicast path fail. You can configure specific OSPF link metrics on topologies to identify paths and build trees to different servers. When a multicast tree gets built with PIM join messages directed toward the source, it follows the most preferred path. A multicast tree to a different multicast source (in a different routing topology) can create another tree along a different path.

Figure 7 on page 55 shows an example of two trees using different paths over different topologies. It shows Server A using the multicast tree with the dashed line as its path and Server B using the multicast tree with the dotted line as its path.

Figure 7: Core Links Configured to Prefer Specified Routing Topologies



RELATED DOCUMENTATION

Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths | 55

Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths

IN THIS SECTION

- Requirements | 56
- Overview | 56
- Configuration | 57
- Verification | 80

This example shows how to use multitopology routing (MTR) to provide redundancy for multicast traffic over separate network paths. That is, two multicast sources send the same multicast stream, yet for redundancy purposes in the case of link failure, the two streams use disjoint paths.

NOTE: Note there is no standard defined at this time for using MTR extensions to PIM.

Requirements

This example requires that Junos OS Release 9.0 or later is running on the provider core devices.

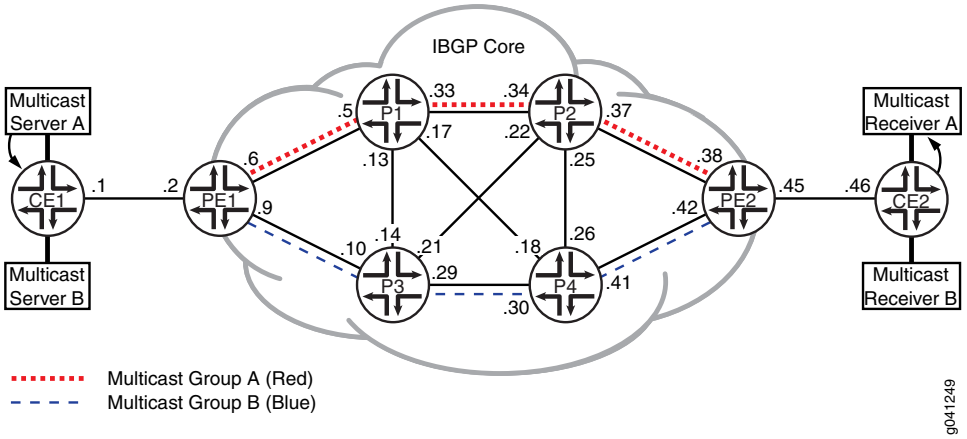
Overview

Assume that each source providing redundant multicast streams, S1 and S2, have different IP subnet addresses. Each source sends multicast traffic using different groups: G1 and G2. Further, assume that S1 and S2 are attached to the same customer edge (CE) device and use BGP to announce routes to the provider edge (PE) router.

For a complete set of configurations for all of the devices in the topology, see "CLI Quick Configuration" on page 57. The remainder of the example focuses on Device CE1 and Device PE1.

Figure 8 on page 56 shows the sample topology.

Figure 8: Multitopology OSPF and BGP for Designating Links Belonging to Voice and Video Services



Configuration

IN THIS SECTION

- Configuring Device CE1 | 66
- Configuring Device PE1 | 71

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 CE1

```
set interfaces fe-0/1/0 fastether-options loopback
set interfaces fe-0/1/0 unit 0 family inet address 11.19.130.1/24
set interfaces fe-0/1/0 unit 0 family inet address 11.19.131.1/24
set interfaces fe-0/1/0 unit 0 family inet address 11.19.132.1/24
set interfaces ge-1/2/0 unit 1 description to-PE1
set interfaces ge-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 97 family inet address 10.255.165.97/32 primary
set protocols bgp group ebgp type external
set protocols bgp group ebgp local-address 10.0.0.1
set protocols bgp group ebgp family inet unicast
set protocols bgp group ebgp family inet multicast
set protocols bgp group ebgp export set_community
set protocols bgp group ebgp export inject_directs
set protocols bgp group ebgp peer-as 100
set protocols bgp group ebgp neighbor 10.0.0.2
set protocols pim interface fe-0/1/0.0 mode sparse
set protocols pim interface ge-1/2/0.1 mode sparse
set policy-options policy-statement inject_directs term a from protocol direct
set policy-options policy-statement inject_directs term a from interface fe-0/1/0.0
set policy-options policy-statement inject_directs term a then next policy
set policy-options policy-statement inject_directs term a then accept
set policy-options policy-statement inject_directs term b then reject
set policy-options policy-statement set_community term a from route-filter 11.19.130.0/24 exact
set policy-options policy-statement set_community term a from route-filter 11.19.131.0/24 exact
set policy-options policy-statement set_community term a then community add red
```

```
set policy-options policy-statement set_community term a then accept
set policy-options policy-statement set_community term b from route-filter 11.19.132.0/24 exact
set policy-options policy-statement set_community term b from route-filter 11.19.133.0/24 exact
set policy-options policy-statement set_community term b then community add blue
set policy-options policy-statement set_community term b then accept
set policy-options policy-statement set_community term default then accept
set policy-options community blue members target:50:50
set policy-options community red members target:40:40
set routing-options interface-routes rib-group inet if-rib
set routing-options static route 10.0.0.0/16 next-hop 10.0.0.2
set routing-options rib-groups inet.2 import-rib inet.0
set routing-options rib-groups if-rib import-rib inet.0
set routing-options rib-groups if-rib import-rib inet.2
set routing-options rib-groups if-rib import-policy inject_directs
set routing-options autonomous-system 101
```

Device CE2

```
set interfaces fe-0/1/1 unit 0
set interfaces ge-1/2/0 unit 46 description to-PE2
set interfaces ge-1/2/0 unit 46 family inet address 10.0.0.46/30
set interfaces lo0 unit 20 family inet address 10.255.165.20/32 primary
set protocols bgp group ebgp type external
set protocols bgp group ebgp local-address 10.0.0.46
set protocols bgp group ebgp peer-as 100
set protocols bgp group ebgp neighbor 10.0.0.45
set routing-options autonomous-system 102
```

Device PE1

```
set interfaces ge-1/2/0 unit 2 description to-CE1
set interfaces ge-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces ge-1/2/1 unit 6 description to-P1
set interfaces ge-1/2/1 unit 6 family inet address 10.0.0.6/30
set interfaces ge-1/2/2 unit 9 description to-P3
set interfaces ge-1/2/2 unit 9 family inet address 10.0.0.9/30
set interfaces lo0 unit 93 family inet address 10.255.165.93/32 primary
```

```
set interfaces lo0 unit 93 family inet address 1.1.1.30/32
set interfaces lo0 unit 93 family inet address 2.2.2.30/32
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.93
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet multicast
set protocols bgp group ibgp export nhs_test
set protocols bgp group ibgp export nhs_inet0_self
set protocols bgp group ibgp neighbor 10.255.165.111
set protocols bgp group ibgp neighbor 10.255.165.203
set protocols bgp group ibgp neighbor 10.255.165.113
set protocols bgp group ibgp neighbor 10.255.165.95
set protocols bgp group ibgp neighbor 10.255.165.99
set protocols bgp group ebgp type external
set protocols bgp group ebgp local-address 10.0.0.2
set protocols bgp group ebgp family inet unicast
set protocols bgp group ebgp family inet multicast
set protocols bgp group ebgp peer-as 101
set protocols bgp group ebgp neighbor 10.0.0.1
set protocols ospf topology red topology-id 126
set protocols ospf topology blue topology-id 52
set protocols ospf area 0.0.0.0 interface ge-1/2/1.6 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/1.6 topology blue metric 1
set protocols ospf area 0.0.0.0 interface ge-1/2/1.6 topology red
set protocols ospf area 0.0.0.0 interface ge-1/2/2.9 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/2.9 topology red metric 1
set protocols ospf area 0.0.0.0 interface ge-1/2/2.9 topology blue
set protocols ospf area 0.0.0.0 interface lo0.93 passive
set protocols ospf area 0.0.0.0 interface 1.1.1.30 topology red
set protocols ospf area 0.0.0.0 interface 1.1.1.30 topology blue disable
set protocols ospf area 0.0.0.0 interface 2.2.2.30 topology blue
set protocols ospf area 0.0.0.0 interface 2.2.2.30 topology red disable
set protocols pim rib-group inet mcast-rib
set protocols pim interface ge-1/2/0.2 mode sparse
set protocols pim interface ge-1/2/1.6 mode sparse
set protocols pim interface ge-1/2/2.9 mode sparse
set policy-options policy-statement nhs_inet0_self term a from protocol bgp
set policy-options policy-statement nhs_inet0_self term a from rib inet.0
set policy-options policy-statement nhs_inet0_self term a then next-hop self
set policy-options policy-statement nhs_test term a from protocol bgp
set policy-options policy-statement nhs_test term a from community red
set policy-options policy-statement nhs_test term a then next-hop 1.1.1.30
```

```
set policy-options policy-statement nhs_test term a then next policy
set policy-options policy-statement nhs_test term a then accept
set policy-options policy-statement nhs_test term b from protocol bgp
set policy-options policy-statement nhs_test term b from community blue
set policy-options policy-statement nhs_test term b then next-hop 2.2.2.30
set policy-options policy-statement nhs_test term b then next policy
set policy-options policy-statement nhs_test term b then accept
set policy-options policy-statement nhs_test term c then next-hop self
set policy-options community blue members target:50:50
set policy-options community red members target:40:40
set routing-options rib-groups mcast-rib import-rib inet.2
set routing-options autonomous-system 100
set routing-options resolution rib inet.2 resolution-ribs :red.inet.0
set routing-options resolution rib inet.2 resolution-ribs :blue.inet.0
set routing-options topologies family inet topology red
set routing-options topologies family inet topology blue
```

Device PE2

```
set interfaces ge-1/2/0 unit 38 description to-P2
set interfaces ge-1/2/0 unit 38 family inet address 10.0.0.38/30
set interfaces ge-1/2/1 unit 42 description to-P4
set interfaces ge-1/2/1 unit 42 family inet address 10.0.0.42/30
set interfaces ge-1/2/2 unit 45 description to-CE2
set interfaces ge-1/2/2 unit 45 family inet address 10.0.0.45/30
set interfaces lo0 unit 203 family inet address 10.255.165.203/32 primary
set interfaces lo0 unit 203 family inet address 1.1.1.40/32
set interfaces lo0 unit 203 family inet address 2.2.2.40/32
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.203
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet multicast
set protocols bgp group ibgp export nhs_test
set protocols bgp group ibgp export nhs_inet0_self
set protocols bgp group ibgp neighbor 10.255.165.111
set protocols bgp group ibgp neighbor 10.255.165.93
set protocols bgp group ibgp neighbor 10.255.165.113
set protocols bgp group ibgp neighbor 10.255.165.95
set protocols bgp group ibgp neighbor 10.255.165.99
set protocols bgp group ebgp type external
```

```
set protocols bgp group ebgp local-address 10.0.0.45
set protocols bgp group ebgp family inet unicast
set protocols bgp group ebgp family inet multicast
set protocols bgp group ebgp peer-as 102
set protocols bgp group ebgp neighbor 10.0.0.46
set protocols ospf topology red topology-id 126
set protocols ospf topology blue topology-id 52
set protocols ospf area 0.0.0.0 interface ge-1/2/0.38 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/0.38 topology blue metric 1
set protocols ospf area 0.0.0.0 interface ge-1/2/0.38 topology red
set protocols ospf area 0.0.0.0 interface ge-1/2/1.42 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/1.42 topology red metric 1
set protocols ospf area 0.0.0.0 interface ge-1/2/1.42 topology blue
set protocols ospf area 0.0.0.0 interface lo0.203 passive
set protocols ospf area 0.0.0.0 interface 1.1.1.40 topology red
set protocols ospf area 0.0.0.0 interface 1.1.1.40 topology blue disable
set protocols ospf area 0.0.0.0 interface 2.2.2.40 topology red disable
set protocols ospf area 0.0.0.0 interface 2.2.2.40 topology blue
set protocols pim rib-group inet mcast-rib
set protocols pim interface ge-1/2/0.38 mode sparse
set protocols pim interface ge-1/2/1.42 mode sparse
set protocols pim interface ge-1/2/2.45 mode sparse
set policy-options policy-statement nhs then next-hop self
set policy-options policy-statement nhs_inet0_self term a from protocol bgp
set policy-options policy-statement nhs_inet0_self term a from rib inet.0
set policy-options policy-statement nhs_inet0_self term a then next-hop self
set policy-options policy-statement nhs_test term a from protocol bgp
set policy-options policy-statement nhs_test term a from community red
set policy-options policy-statement nhs_test term a then next-hop 1.1.1.40
set policy-options policy-statement nhs_test term a then next policy
set policy-options policy-statement nhs_test term a then accept
set policy-options policy-statement nhs_test term b from protocol bgp
set policy-options policy-statement nhs_test term b from community blue
set policy-options policy-statement nhs_test term b then next-hop 2.2.2.40
set policy-options policy-statement nhs_test term b then next policy
set policy-options policy-statement nhs_test term b then accept
set policy-options policy-statement nhs_test term c then next-hop self
set policy-options community blue members target:50:50
set policy-options community red members target:40:40
set routing-options rib-groups mcast-rib import-rib inet.2
set routing-options autonomous-system 100
set routing-options resolution rib inet.2 resolution-ribs :red.inet.0
```

```
set routing-options resolution rib inet.2 resolution-ribs :blue.inet.0
set routing-options topologies family inet topology red
set routing-options topologies family inet topology blue
```

Device P1

```
set interfaces ge-1/2/0 unit 5 description to-PE1
set interfaces ge-1/2/0 unit 5 family inet address 10.0.0.5/30
set interfaces ge-1/2/1 unit 13 description to-P3
set interfaces ge-1/2/1 unit 13 family inet address 10.0.0.13/30
set interfaces ge-1/2/2 unit 17 description to-P4
set interfaces ge-1/2/2 unit 17 family inet address 10.0.0.17/30
set interfaces ge-1/2/3 unit 33 description to-P2
set interfaces ge-1/2/3 unit 33 family inet address 10.0.0.33/30
set interfaces lo0 unit 99 family inet address 10.255.165.99/32 primary
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.99
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet multicast
set protocols bgp group ibgp neighbor 10.255.165.93
set protocols bgp group ibgp neighbor 10.255.165.111
set protocols bgp group ibgp neighbor 10.255.165.203
set protocols bgp group ibgp neighbor 10.255.165.95
set protocols bgp group ibgp neighbor 10.255.165.113
set protocols ospf topology red topology-id 126
set protocols ospf topology blue topology-id 52
set protocols ospf area 0.0.0.0 interface ge-1/2/3.33 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/3.33 topology red
set protocols ospf area 0.0.0.0 interface ge-1/2/3.33 topology blue metric 1
set protocols ospf area 0.0.0.0 interface ge-1/2/2.17
set protocols ospf area 0.0.0.0 interface ge-1/2/1.13
set protocols ospf area 0.0.0.0 interface ge-1/2/0.5 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/0.5 topology red
set protocols ospf area 0.0.0.0 interface ge-1/2/0.5 topology blue metric 1
set protocols ospf area 0.0.0.0 interface lo0.99 passive
set protocols pim rib-group inet mcast-rib
set protocols pim interface ge-1/2/3.33 mode sparse
set protocols pim interface ge-1/2/2.17 mode sparse
set protocols pim interface ge-1/2/1.13 mode sparse
set protocols pim interface ge-1/2/0.5 mode sparse
```

```
set routing-options rib-groups mcast-rib import-rib inet.2
set routing-options autonomous-system 100
set routing-options topologies family inet topology red
set routing-options topologies family inet topology blue
```

Device P2

```
set interfaces ge-1/2/0 unit 22 description to-P3
set interfaces ge-1/2/0 unit 22 family inet address 10.0.0.22/30
set interfaces ge-1/2/1 unit 25 description to-P4
set interfaces ge-1/2/1 unit 25 family inet address 10.0.0.25/30
set interfaces ge-1/2/2 unit 34 description to-P1
set interfaces ge-1/2/2 unit 34 family inet address 10.0.0.34/30
set interfaces ge-1/2/3 unit 37 description to-PE2
set interfaces ge-1/2/3 unit 37 family inet address 10.0.0.37/30
set interfaces lo0 unit 113 family inet address 10.255.165.113/32 primary
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.113
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet multicast
set protocols bgp group ibgp neighbor 10.255.165.93
set protocols bgp group ibgp neighbor 10.255.165.111
set protocols bgp group ibgp neighbor 10.255.165.203
set protocols bgp group ibgp neighbor 10.255.165.95
set protocols bgp group ibgp neighbor 10.255.165.99
set protocols ospf topology red topology-id 126
set protocols ospf topology blue topology-id 52
set protocols ospf area 0.0.0.0 interface ge-1/2/2.34 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/2.34 topology red
set protocols ospf area 0.0.0.0 interface ge-1/2/2.34 topology blue metric 1
set protocols ospf area 0.0.0.0 interface ge-1/2/0.22
set protocols ospf area 0.0.0.0 interface ge-1/2/1.25
set protocols ospf area 0.0.0.0 interface ge-1/2/3.37 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/3.37 topology red
set protocols ospf area 0.0.0.0 interface ge-1/2/3.37 topology blue metric 1
set protocols ospf area 0.0.0.0 interface lo0.113 passive
set protocols pim rib-group inet mcast-rib
set protocols pim interface ge-1/2/2.34 mode sparse
set protocols pim interface ge-1/2/0.22 mode sparse
set protocols pim interface ge-1/2/1.25 mode sparse
```

```
set protocols pim interface ge-1/2/3.37 mode sparse
set routing-options rib-groups mcast-rib import-rib inet.2
set routing-options autonomous-system 100
set routing-options topologies family inet topology red
set routing-options topologies family inet topology blue
```

Device P3

```
set interfaces ge-1/2/0 unit 10 description to-PE1
set interfaces ge-1/2/0 unit 10 family inet address 10.0.0.10/30
set interfaces ge-1/2/1 unit 14 description to-P1
set interfaces ge-1/2/1 unit 14 family inet address 10.0.0.14/30
set interfaces ge-1/2/2 unit 21 description to-P2
set interfaces ge-1/2/2 unit 21 family inet address 10.0.0.21/30
set interfaces ge-1/2/3 unit 29 description to-P4
set interfaces ge-1/2/3 unit 29 family inet address 10.0.0.29/30
set interfaces lo0 unit 111 family inet address 10.255.165.111/32 primary
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.111
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet multicast
set protocols bgp group ibgp neighbor 10.255.165.93
set protocols bgp group ibgp neighbor 10.255.165.113
set protocols bgp group ibgp neighbor 10.255.165.203
set protocols bgp group ibgp neighbor 10.255.165.95
set protocols bgp group ibgp neighbor 10.255.165.99
set protocols ospf topology red topology-id 126
set protocols ospf topology blue topology-id 52
set protocols ospf area 0.0.0.0 interface ge-1/2/3.29 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/3.29 topology red metric 1
set protocols ospf area 0.0.0.0 interface ge-1/2/3.29 topology blue
set protocols ospf area 0.0.0.0 interface ge-1/2/2.21
set protocols ospf area 0.0.0.0 interface ge-1/2/1.14
set protocols ospf area 0.0.0.0 interface ge-1/2/0.10 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/0.10 topology red metric 1
set protocols ospf area 0.0.0.0 interface ge-1/2/0.10 topology blue
set protocols ospf area 0.0.0.0 interface lo0.111 passive
set protocols pim rib-group inet mcast-rib
set protocols pim interface ge-1/2/3.29 mode sparse
set protocols pim interface ge-1/2/2.21 mode sparse
```

```
set protocols pim interface ge-1/2/1.14 mode sparse
set protocols pim interface ge-1/2/0.10 mode sparse
set routing-options rib-groups mcast-rib import-rib inet.2
set routing-options autonomous-system 100
set routing-options topologies family inet topology red
set routing-options topologies family inet topology blue
```

Device P4

```
set interfaces ge-1/2/0 unit 18 description to-P1
set interfaces ge-1/2/0 unit 18 family inet address 10.0.0.18/30
set interfaces ge-1/2/1 unit 26 description to-P2
set interfaces ge-1/2/1 unit 26 family inet address 10.0.0.26/30
set interfaces ge-1/2/2 unit 30 description to-P3
set interfaces ge-1/2/2 unit 30 family inet address 10.0.0.30/30
set interfaces ge-1/2/3 unit 41 description to-PE2
set interfaces ge-1/2/3 unit 41 family inet address 10.0.0.41/30
set interfaces lo0 unit 95 family inet address 10.255.165.95/32 primary
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.165.95
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet multicast
set protocols bgp group ibgp neighbor 10.255.165.93
set protocols bgp group ibgp neighbor 10.255.165.113
set protocols bgp group ibgp neighbor 10.255.165.203
set protocols bgp group ibgp neighbor 10.255.165.111
set protocols bgp group ibgp neighbor 10.255.165.99
set protocols ospf topology red topology-id 126
set protocols ospf topology blue topology-id 52
set protocols ospf area 0.0.0.0 interface ge-1/2/2.30 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/2.30 topology red metric 1
set protocols ospf area 0.0.0.0 interface ge-1/2/2.30 topology blue
set protocols ospf area 0.0.0.0 interface ge-1/2/0.18
set protocols ospf area 0.0.0.0 interface ge-1/2/1.26
set protocols ospf area 0.0.0.0 interface ge-1/2/3.41 metric 10
set protocols ospf area 0.0.0.0 interface ge-1/2/3.41 topology red metric 1
set protocols ospf area 0.0.0.0 interface ge-1/2/3.41 topology blue
set protocols ospf area 0.0.0.0 interface lo0.95 passive
set protocols pim rib-group inet mcast-rib
set protocols pim interface ge-1/2/2.30 mode sparse
```

```
set protocols pim interface ge-1/2/0.18 mode sparse
set protocols pim interface ge-1/2/1.26 mode sparse
set protocols pim interface ge-1/2/3.41 mode sparse
set routing-options rib-groups mcast-rib import-rib inet.2
set routing-options autonomous-system 100
set routing-options topologies family inet topology red
set routing-options topologies family inet topology blue
```

Configuring Device CE1

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 CE1:

1. Configure the interfaces.

   For demonstration purposes, the example places an Ethernet interface into loopback mode and configures several addresses on this loopback interface. The addresses are then announced to the network as direct routes. These routes simulate a group of BGP routes with communities attached.

   ```
   [edit interfaces]
   user@CE1# set fe-0/1/0 fastether-options loopback
   user@CE1# set fe-0/1/0 unit 0 family inet address 11.19.130.1/24
   user@CE1# set fe-0/1/0 unit 0 family inet address 11.19.131.1/24
   user@CE1# set fe-0/1/0 unit 0 family inet address 11.19.132.1/24
   user@CE1# set ge-1/2/0 unit 1 description to-PE1
   user@CE1# set ge-1/2/0 unit 1 family inet address 10.0.0.1/30
   user@CE1# set lo0 unit 97 family inet address 10.255.165.97/32 primary
   ```

2. Configure the external BGP (EBGP) connection to Device PE1.

   The CE router nearest to the multicast servers announces the multicast source IP addresses to the PE routers using EBGP. The source addresses are announced with both **family inet unicast** and **family inet multicast**, thus causing the BGP route to be added to the default routing table, **inet.0**, and to the multicast routing table, **inet.2**. Both sets of routes are injected by the PE router into IBGP.

   ```
   [edit protocols bgp group ebgp]
   user@CE1# set type external
   user@CE1# set local-address 10.0.0.1
   ```

```
user@CE1# set family inet unicast
user@CE1# set family inet multicast
user@CE1# set peer-as 100
user@CE1# set neighbor 10.0.0.2
```

3. Configure PIM on the interfaces.

```
[edit protocols pim]
user@CE1# set interface fe-0/1/0.0 mode sparse
user@CE1# set interface ge-1/2/0.1 mode sparse
```

4. Configure the routing policy that announces the addresses that are configured on interface fe-0/1/0.

```
[edit policy-options policy-statement inject_directs]
user@CE1# set term a from protocol direct
user@CE1# set term a from interface fe-0/1/0.0
user@CE1# set term a then next policy
user@CE1# set term a then accept
user@CE1# set term b then reject
```

5. Configure the routing policy that tags some routes with the red community attribute and other routes with the blue community attribute.

   The CE router advertises routes through EBGP to the PE router. These routes are advertised as BGP **family inet multicast** routes with communities set for two different groups. Policies identify the two groups of BGP routes.

```
[edit policy-options policy-statement set_community term a]
user@CE1# set from route-filter 11.19.130.0/24 exact
user@CE1# set from route-filter 11.19.131.0/24 exact
user@CE1# set then community add red
user@CE1# set then accept
[edit policy-options policy-statement set_community term b]
user@CE1# set from route-filter 11.19.132.0/24 exact
user@CE1# set from route-filter 11.19.133.0/24 exact
user@CE1# set then community add blue
user@CE1# set then accept
[edit policy-options policy-statement set_community term default]
user@CE1# set then accept
[edit policy-options]
user@CE1# set community blue members target:50:50
```

```
user@CE1# set community red members target:40:40
```

6. Apply the **set_community** export policy so that the direct routes are exported into BGP.

   Apply the **inject_directs** export policy to announce the addresses that are configured on interface fe-0/1/0.

```
[edit protocols bgp group ebgp]
user@CE1# set export set_community
user@CE1# set export inject_directs
```

7. Use **rib-groups** to simulate a group of BGP routes with communities attached and announced as multicast routes.

   This configuration creates a multicast routing table and causes PIM to use the multicast routing table **inet.2**.

```
[edit routing-options]
user@CE1# set interface-routes rib-group inet if-rib
user@CE1# set static route 10.0.0.0/16 next-hop 10.0.0.2
[edit routing-options rib-groups]
user@CE1# set inet.2 import-rib inet.0
user@CE1# set if-rib import-rib inet.0
user@CE1# set if-rib import-rib inet.2
user@CE1# set if-rib import-policy inject_directs
```

8. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@CE1# set autonomous-system 101
```

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@CE1# show interfaces
fe-0/1/0 {
    fastether-options {
        loopback;
```

```
    }
    unit 0 {
        family inet {
            address 11.19.130.1/24;
            address 11.19.131.1/24;
            address 11.19.132.1/24;
        }
    }
}
ge-1/2/0 {
    unit 1 {
        description to-PE1;
        family inet {
            address 10.0.0.1/30;
        }
    }
}
lo0 {
    unit 97 {
        family inet {
            address 10.255.165.97/32 {
                primary;
            }
        }
    }
}
```

```
user@CE1# show protocols
bgp {
    group ebgp {
        type external;
        local-address 10.0.0.1;
        family inet {
            unicast;
            multicast;
        }
        export [ set_community inject_directs ];
        peer-as 100;
        neighbor 10.0.0.2;
    }
}
pim {
    interface fe-0/1/0.0 {
        mode sparse;
```

```
    }
    interface ge-1/2/0.1 {
        mode sparse;
    }
}
```

```
user@CE1# show policy-options
policy-statement inject_directs {
    term a {
        from {
            protocol direct;
            interface fe-0/1/0.0;
        }
        then {
            next policy;
            accept;
        }
    }
    term b {
        then reject;
    }
}
policy-statement set_community {
    term a {
        from {
            route-filter 11.19.130.0/24 exact;
            route-filter 11.19.131.0/24 exact;
        }
        then {
            community add red;
            accept;
        }
    }
    term b {
        from {
            route-filter 11.19.132.0/24 exact;
            route-filter 11.19.133.0/24 exact;
        }
        then {
            community add blue;
            accept;
        }
    }
    term default {
```

```
        then accept;
    }
}
community blue members target:50:50;
community red members target:40:40;
```

```
user@CE1# show routing-options
interface-routes {
    rib-group inet if-rib;
}
static {
    route 10.0.0.0/16 next-hop 10.0.0.2;
}
rib-groups {
    inet.2 {
        import-rib inet.0;
    }
    if-rib {
        import-rib [ inet.0 inet.2 ];
        import-policy inject_directs;
    }
}
autonomous-system 101;
```

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

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 ge-1/2/0 unit 2 description to-CE1
user@PE1# set ge-1/2/0 unit 2 family inet address 10.0.0.2/30
user@PE1# set ge-1/2/1 unit 6 description to-P1
user@PE1# set ge-1/2/1 unit 6 family inet address 10.0.0.6/30
user@PE1# set ge-1/2/2 unit 9 description to-P3
user@PE1# set ge-1/2/2 unit 9 family inet address 10.0.0.9/30
```

```
user@PE1# set lo0 unit 93 family inet address 10.255.165.93/32 primary
```

2. Configure secondary addresses, 1.1.1.30 and 2.2.2.30.

   A specific protocol next-hop IP address is required for each topology on each router injecting IBGP routes. You can configure multiple secondary loopback IP addresses on a router to be used as protocol next-hop addresses. This configuration shows nonprimary IP addresses 1.1.1.30/32 and 2.2.2.30/32 configured on loopback interface lo0 for use in the red and blue topologies, respectively.

   A group of BGP routes associated with a routing topology use the same unique protocol next hop. For instance, if you configure a PE router to handle two routing topologies, then you would also configure two unique nonprimary addresses under loopback interface lo0.

```
[edit interfaces]
user@PE1# set lo0 unit 93 family inet address 1.1.1.30/32
user@PE1# set lo0 unit 93 family inet address 2.2.2.30/32
```

3. Associate each nonprimary loopback IP address with a topology for inclusion in the associated topology routing table.

   Configure the loopback IP address and topology under an OSPF interface statement. You must specifically disable all other topologies known to OSPF for two reasons. First, the loopback address specific to a topology must reside in only one topology routing table. Second, once the topology is added to OSPF, the topology defaults to being enabled on all subsequent interfaces under OSPF.

   The Device PE1 configuration places the loopback address 1.1.1.30/32 into the OSPF database as a stub route under this router's OSPF Router-LSA. It belongs to the red and default topologies, but not to the blue topology. The loopback address 1.1.1.30/32 is installed in the remote core routers' topology routing tables **inet.0** and **:red.inet.0**, (but not in **:blue.inet.0**). Use a similar configuration for the blue loopback address 2.2.2.30/32.

```
[edit protocols ospf]
user@PE1# set topology red topology-id 126
user@PE1# set topology blue topology-id 52
[edit protocols ospf area 0.0.0.0]
user@PE1# set interface 1.1.1.30 topology red
user@PE1# set interface 1.1.1.30 topology blue disable
user@PE1# set interface 2.2.2.30 topology blue
user@PE1# set interface 2.2.2.30 topology red disable
```

4. Enable OSPF on the interfaces, and configure specific OSPF link metrics on topologies to identify paths and build trees to different servers.

Links can support all routing topologies to provide backup should a primary multicast path fail.

When a multicast tree gets built through PIM join messages directed toward the source, it follows the most preferred path. A multicast tree to a different multicast source (in a different routing topology) can create another tree along a different path.

```
[edit protocols ospf area 0.0.0.0]
user@PE1# set interface ge-1/2/1.6 metric 10
user@PE1# set interface ge-1/2/1.6 topology blue metric 1
user@PE1# set interface ge-1/2/1.6 topology red
user@PE1# set interface ge-1/2/2.9 metric 10
user@PE1# set interface ge-1/2/2.9 topology red metric 1
user@PE1# set interface ge-1/2/2.9 topology blue
user@PE1# set interface lo0.93 passive
```

5. Create the multicast routing table **inet.2**, and configure PIM to use the **inet.2** routing table.

   Set up a separate routing table for multicast lookups. It is populated with routes from **inet.2**. The **inet.2** routing table is populated by routes of type multicast.

   ```
   [edit routing-options]
   user@PE1# set rib-groups mcast-rib import-rib inet.2
   ```

6. Configure PIM to use the routes in **inet.2**.

   ```
   [edit protocols pim]
   user@PE1# set rib-group inet mcast-rib
   ```

7. Enable PIM on the interfaces.

   ```
   [edit protocols pim]
   user@PE1# set interface ge-1/2/0.2 mode sparse
   user@PE1# set interface ge-1/2/1.6 mode sparse
   user@PE1# set interface ge-1/2/2.9 mode sparse
   ```

8. Configure the router to perform route resolution on protocol next hops using specified routing tables.

   The protocol next hop is used to determine the forwarding next-hop interface out of which to forward PIM join messages. This configuration directs **inet.2** route resolution to use topology routing tables **:red.inet.0** and **:blue.inet.0** for protocol next-hop IP address lookups.

   You can specify up to two routing tables in the resolution configuration. A key element to this solution is that the protocol next-hop address resides in only one topology routing table. That is, the protocol

next hop belongs to a remote PE secondary loopback address and is injected into only one topology routing table. The route resolution scheme first checks routing table **:red.inet.0** for the protocol next-hop address. If the address is found, it uses this entry. If it is not found, the resolution scheme checks routing table **:blue.inet.0**. Hence, only one topology routing table is used for each protocol nexthop address.

```
[edit routing-options resolution rib inet.2]
user@PE1# set resolution-ribs :red.inet.0
user@PE1# set resolution-ribs :blue.inet.0
```

9. Configure the autonomous system (AS) number.

```
[edit routing-options]
user@PE1# set autonomous-system 100
```

10. Configure BGP.

```
[edit protocols bgp group ibgp]
user@PE1# set type internal
user@PE1# set local-address 10.255.165.93
user@PE1# set family inet unicast
user@PE1# set family inet multicast
user@PE1# set neighbor 10.255.165.111
user@PE1# set neighbor 10.255.165.203
user@PE1# set neighbor 10.255.165.113
user@PE1# set neighbor 10.255.165.95
user@PE1# set neighbor 10.255.165.99
[edit protocols bgp group ebgp]
user@PE1# set type external
user@PE1# set local-address 10.0.0.2
user@PE1# set family inet unicast
user@PE1# set family inet multicast
user@PE1# set peer-as 101
user@PE1# set neighbor 10.0.0.1
```

11. Set the protocol next hop when exporting EBGP routes into IBGP.

    Configure the ingress Device PE1 router to set the BGP route's protocol next-hop address when exporting the route into IBGP.

    BGP uses an export policy to set the next hop when injecting the EBGP routes into IBGP.

This configuration is an export policy where there are three possibilities of next hops being set. Route 1.1.1.30 is associated with the red topology. Route 2.2.2.30 is associated with the blue topology. For the default next-hop self policy, the primary loopback address 10.255.165.93 on Device PE1 is used.

The **nhs_test** policy sets the protocol next-hop based on the community in the BGP update.

```
[edit policy-options]
user@PE1# set community blue members target:50:50
user@PE1# set community red members target:40:40
[edit policy-options policy-statement nhs_test term a]
user@PE1# set from protocol bgp
user@PE1# set from community red
user@PE1# set then next-hop 1.1.1.30
user@PE1# set then next policy
user@PE1# set then accept
[edit policy-options policy-statement nhs_test term b]
user@PE1# set from protocol bgp
user@PE1# set from community blue
user@PE1# set then next-hop 2.2.2.30
user@PE1# set then next policy
user@PE1# set then accept
user@PE1# set policy-options policy-statement nhs_test term c then next-hop self
[edit policy-options policy-statement nhs_inet0_self term a]
user@PE1# set from protocol bgp
user@PE1# set from rib inet.0
user@PE1# set then next-hop self
```

12. Apply the next-hop self policies to the IBGP sessions.

```
[edit protocols bgp group ibgp]
user@PE1# set export nhs_test
user@PE1# set export nhs_inet0_self
```

13. Configure the voice and video topologies, which enable you to use these topologies with OSPF and BGP.

    The names **voice** and **video** are local to the router. The names are not propagated beyond this router. However, for management purposes, a consistent naming scheme across routers in a multitopology environment is convenient.

```
[edit routing-options topologies family inet]
user@PE1# set topology red
user@PE1# set topology blue
```

Results

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

```
user@PE1# show interfaces
ge-1/2/0 {
    unit 2 {
        description to-CE1;
        family inet {
            address 10.0.0.2/30;
        }
    }
}
ge-1/2/1 {
    unit 6 {
        description to-P1;
        family inet {
            address 10.0.0.6/30;
        }
    }
}
ge-1/2/2 {
    unit 9 {
        description to-P3;
        family inet {
            address 10.0.0.9/30;
        }
    }
}
lo0 {
    unit 93 {
        family inet {
            address 10.255.165.93/32 {
                primary;
            }
            address 1.1.1.30/32;
            address 2.2.2.30/32;
        }
    }
}
```

```
user@PE1# show protocols
```

```
bgp {
    group ibgp {
        type internal;
        local-address 10.255.165.93;
        family inet {
            unicast;
            multicast;
        }
        export [ nhs_test nhs_inet0_self ];
        neighbor 10.255.165.111;
        neighbor 10.255.165.203;
        neighbor 10.255.165.113;
        neighbor 10.255.165.95;
        neighbor 10.255.165.99;
    }
    group ebgp {
        type external;
        local-address 10.0.0.2;
        family inet {
            unicast;
            multicast;
        }
        peer-as 101;
        neighbor 10.0.0.1;
    }
}
ospf {
    topology red topology-id 126;
    topology blue topology-id 52;
    area 0.0.0.0 {
        interface ge-1/2/1.6 {
            metric 10;
            topology blue metric 1;
            topology red;
        }
        interface ge-1/2/2.9 {
            metric 10;
            topology red metric 1;
            topology blue;
        }
        interface lo0.93 {
            passive;
        }
        interface 1.1.1.30 {
```

```
            topology red;
            topology blue disable;
        }
        interface 2.2.2.30 {
            topology blue;
            topology red disable;
        }
    }
}
pim {
    rib-group inet mcast-rib;
    interface ge-1/2/0.2 {
        mode sparse;
    }
    interface ge-1/2/1.6 {
        mode sparse;
    }
    interface ge-1/2/2.9 {
        mode sparse;
    }
}
```

```
user@PE1# show policy-options
policy-statement nhs_inet0_self {
    term a {
        from {
            protocol bgp;
            rib inet.0;
        }
        then {
            next-hop self;
        }
    }
}
policy-statement nhs_test {
    term a {
        from {
            protocol bgp;
            community red;
        }
        then {
            next-hop 1.1.1.30;
            next policy;
            accept;
```

```
        }
    }
    term b {
        from {
            protocol bgp;
            community blue;
        }
        then {
            next-hop 2.2.2.30;
            next policy;
            accept;
        }
    }
    term c {
        then {
            next-hop self;
        }
    }
}
community blue members target:50:50;
community red members target:40:40;
```

```
user@PE1# show routing-options
rib-groups {
    mcast-rib {
        import-rib inet.2;
    }
}
autonomous-system 100;
resolution {
    rib inet.2 {
        resolution-ribs [ :red.inet.0 :blue.inet.0 ];
    }
}
topologies {
    family inet {
        topology red;
        topology blue;
    }
}
```

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

Verification

IN THIS SECTION

- Checking the IBGP routes in inet.2 | 80
- Verifying the Routes | 81
- Checking the Resolving BGP Next Hops | 83
- Examining the Protocol Next Hop | 84
- Verifying the OSPF Neighbor | 85
- Checking the Router LSA | 86
- Checking How Traffic Traverses the Network | 87

Confirm that the configuration is working properly.

Checking the IBGP routes in inet.2

Purpose

Make sure that the routes injected into IBGP by Device PE1 have next hops that are based on the topology to which they belong.

Action

From operational mode, enter the **show route table extensive** command.

```
user@PE1> show route 11.19.130.0/24 table inet.2 extensive
```

```
inet.2: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
11.19.130.0/24 (1 entry, 1 announced)
TSI:
Page 0 idx 0 Type 1 val 93e9768
    Flags: Nexthop Change
    Nexthop: 1.1.1.30
    Localpref: 100
    AS path: [100] 101 I
    Communities: target:40:40
Path 11.19.130.0 from 10.0.0.1 Vector len 4.  Val: 0
        *BGP    Preference: 170/-101
                Next hop type: Router, Next hop index: 1180
                Address: 0x94003ec
```

```
                Next-hop reference count: 16
                Source: 10.0.0.1
                Next hop: 10.0.0.1 via lt-1/2/0.2, selected
                Session Id: 0x380004
                State: <Active Ext>
                Local AS:   100 Peer AS:   101
                Age: 22
                Validation State: unverified
                Task: BGP_101.10.0.0.1+58346
                Announcement bits (1): 0-BGP_RT_Background
                AS path: 101 I
                Communities: target:40:40
                Accepted
                Localpref: 100
                Router ID: 10.255.165.97
```

Meaning

This output shows an IBGP route in the **inet.2** routing table, as seen from Device PE1. The route was originally injected into IBGP by Device PE1, where the next hop was set based on the topology to which the route belonged. The BGP community value determined the topology association.

The route 11.19.130/24 belongs to the red topology because it has a community value of target:40:40. The protocol next hop is 1.1.1.30, and the forwarding next hop is ge-1/2/1.42.

Verifying the Routes

Purpose

Make sure that the routes are in the expected routing tables and that the expected communities are attached to the routes.

Action

From operational mode, enter the **show route detail** command on Device PE1.

`user@PE1>` **show route 11.19.130.0/24 detail**

```
inet.0: 29 destinations, 30 routes (29 active, 0 holddown, 0 hidden)
11.19.130.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Router, Next hop index: 812
                Address: 0xb9f064c
                Next-hop reference count: 22
```

```
                Source: 10.0.0.1
                Next hop: 10.0.0.1 via fe-1/2/0.2, selected
                Session Id: 0x600004
                State: <Active Ext>
                Local AS:   100 Peer AS:   101
                Age: 3d 21:44:07
                Task: BGP_101.10.0.0.1+51873
                Announcement bits (3): 0-KRT 3-BGP_RT_Background 4-Resolve tree 3

                AS path: 101 I
                Communities: target:40:40
                Accepted
                Localpref: 100
                Router ID: 10.255.165.97
                Secondary Tables: :voice.inet.0

:voice.inet.0: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)

11.19.130.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Router, Next hop index: 812
                Address: 0xb9f064c
                Next-hop reference count: 22
                Source: 10.0.0.1
                Next hop: 10.0.0.1 via fe-1/2/0.2, selected
                Session Id: 0x600004
                State: <Secondary Active IndepResolution Ext>
                Local AS:   100 Peer AS:   101
                Age: 3d 21:44:07
                Task: BGP_101.10.0.0.1+51873
                Announcement bits (2): 0-KRT 1-Resolve tree 1
                AS path: 101 I
                Communities: target:40:40
                Accepted
                Localpref: 100
                Router ID: 10.255.165.97
                Primary Routing Table inet.0
```

Meaning

This output shows BGP route 11.19.130.0/24 with community value target:40:40. Because the route matches the criteria for the voice topology, it is added to both the default and voice topology routing tables (**inet.0** and **:voice.inet.0**). Device PE1 learns the route from Device CE1 through EBGP and then injects the route into IBGP.

Checking the Resolving BGP Next Hops

Purpose

Check the protocol next hop and forwarding next hop.

Action

From operational mode, enter the **show route detail** command on Device PE2.

```
user@PE2> show route 11.19.130.0/24 detail
```

```
inet.0: 29 destinations, 30 routes (29 active, 0 holddown, 0 hidden)
11.19.130.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0xb9f0e04
                Next-hop reference count: 12
                Source: 10.255.165.93
                Next hop type: Router, Next hop index: 262153
                Next hop: 10.0.0.37 via fe-1/2/0.38
                Session Id: 0x700004
                Next hop: 10.0.0.41 via fe-1/2/1.42, selected
                Session Id: 0x700005
                Protocol next hop: 10.255.165.93
                Indirect next hop: bb8c000 262154 INH Session ID: 0x700007
                State: <Active Int Ext>
                Local AS:   100 Peer AS:   100
                Age: 3d 4:27:40         Metric2: 30
                Task: BGP_100.10.255.165.93+179
                Announcement bits (3): 0-KRT 3-BGP_RT_Background 4-Resolve tree 3

                AS path: 101 I
                Communities: target:40:40
                Accepted
                Localpref: 100
                Router ID: 10.255.165.93
                Secondary Tables: :voice.inet.0

:voice.inet.0: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)

11.19.130.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0xb9f0f34
                Next-hop reference count: 6
```

```
                Source: 10.255.165.93
                Next hop type: Router, Next hop index: 1188
                Next hop: 10.0.0.37 via fe-1/2/0.38, selected
                Session Id: 0x700004
                Protocol next hop: 10.255.165.93
                Indirect next hop: bb8c1d8 262177 INH Session ID: 0x700007
                State: <Secondary Active IndepResolution Int Ext>
                Local AS:   100 Peer AS:   100
                Age: 3d 2:00:20         Metric2: 30
                Task: BGP_100.10.255.165.93+179
                Announcement bits (2): 0-KRT 1-Resolve tree 1
                AS path: 101 I
                Communities: target:40:40
                Accepted
                Localpref: 100
                Router ID: 10.255.165.93
                Primary Routing Table inet.0
```

Meaning

A typical IBGP core has BGP routes with protocol next hops that resolve using the underlying IGP routes. IBGP routes in a topology routing table have protocol next-hop IP addresses. By default, the same topology routing table is used to look up and resolve the protocol next-hop IP address to a forwarding next hop. This output from Device PE2 shows the same BGP route as seen in the previous example: 11.19.130.0/24. The route is being shown from a different perspective, that is, from Device PE2 as an IBGP route. Similarly, this IBGP route is added to both **inet.0** and **:voice.inet.0** on Device PE2. However, while each route has the same protocol next hop, each route has a different forwarding next hop (ge-0/0/3.0 instead of ge-0/1/4.0). The reason for this difference is when the protocol next-hop IP address 10.255.165.93 is resolved, it uses the corresponding routing table (**inet.0** or **:voice.inet.0**) to look up the protocol next hop.

Examining the Protocol Next Hop

Purpose

Check the protocol next hop and forwarding next hop.

Action

From operational mode, enter the **show route** command on Device PE2.

`user@PE2>` **show route 10.255.165.93**

```
inet.0: 29 destinations, 30 routes (29 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.255.165.93/32   *[OSPF/10] 3d 04:37:26, metric 30
                    > to 10.0.0.37 via fe-1/2/0.38
                      to 10.0.0.41 via fe-1/2/1.42

:voice.inet.0: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.165.93/32   *[OSPF/10] 3d 02:10:04, metric 30
                    > to 10.0.0.37 via fe-1/2/0.38

:video.inet.0: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.165.93/32   *[OSPF/10] 3d 02:03:16, metric 30
                    > to 10.0.0.41 via fe-1/2/1.42
```

Meaning

This output from Device PE2 shows the protocol next hop of 11.19.130.0/24, which is IP address 10.255.165.93, thus further demonstrating how IBGP route 11.19.130.0/24 resolves its protocol next hop. The forwarding next hops of 10.255.165.93 match the IBGP forwarding next hops of route 11.19.130/24 as shown in the previous example. Observe here that the IP address 10.255.165.93 is also in routing table **:video.inet.0**. This address is the loopback address of Device PE1, and as such, resides in all three routing tables. This example also shows how traffic entering Device PE2 destined to 11.19.130.0/24 exits different interfaces depending on its associated topology. The actual traffic is marked in such a way that a firewall filter can direct the traffic to use a particular topology routing table.

Verifying the OSPF Neighbor

Purpose

Make sure that the expected topologies are enabled on the OSPF neighbor.

Action

From operational mode, enter the **show (ospf | ospf3) neighbor extensive** command on Device P2.

`user@P2>` **show ospf neighbor 10.0.0.21 extensive**

```
Address          Interface              State     ID               Pri  Dead
10.0.0.21        fe-1/2/0.22            Full      10.255.165.111   128    39
  Area 0.0.0.0, opt 0x52, DR 10.0.0.22, BDR 10.0.0.21
  Up 3d 06:09:50, adjacent 3d 06:09:50
   Topology default (ID 0) -> Bidirectional
   Topology video (ID 52) -> Bidirectional
```

Meaning

This Device P2 output shows OSPF neighbor PE2 (10.0.0.21), where multitopology OSPF default and video are participants. The **Bidirectional** flag shows that the neighbor is configured using the same multitopology OSPF ID.

Checking the Router LSA

Purpose

Check the links where video and voice topologies are enabled.

Action

From operational mode, enter the **show ospf database extensive** command on Device P2.

```
user@P2> show ospf database lsa-id 10.255.165.203 extensive
```

```
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.255.165.203   10.255.165.203   0x80000063  1552  0x22 0xdff3  80
  bits 0x0, link count 3
  id 10.255.165.203, data 255.255.255.255, Type Stub (3)
    Topology count: 2, Default metric: 0
    Topology video (ID 52) -> Metric: 0
    Topology voice (ID 126) -> Metric: 0
  id 10.0.0.38, data 10.0.0.38, Type Transit (2)
    Topology count: 2, Default metric: 10
    Topology video (ID 52) -> Metric: 200
    Topology voice (ID 126) -> Metric: 10
  id 10.0.0.42, data 10.0.0.42, Type Transit (2)
    Topology count: 1, Default metric: 10
    Topology video (ID 52) -> Metric: 10
  Topology default (ID 0)
    Type: Transit, Node ID: 10.0.0.42
      Metric: 10, Bidirectional
    Type: Transit, Node ID: 10.0.0.38
      Metric: 10, Bidirectional
  Topology video (ID 52)
    Type: Transit, Node ID: 10.0.0.42
      Metric: 10, Bidirectional
    Type: Transit, Node ID: 10.0.0.38
      Metric: 200, Bidirectional
  Topology voice (ID 126)
    Type: Transit, Node ID: 10.0.0.38
      Metric: 10, Bidirectional
```

```
  Aging timer 00:34:08
  Installed 00:25:49 ago, expires in 00:34:08, sent 00:25:47 ago
  Last changed 3d 01:45:51 ago, Change count: 10
```

Meaning

This Device P2 output shows the Router-LSA originated by Device PE2. The LSA shows links where video and voice topologies are enabled (in addition to the default topology).

Checking How Traffic Traverses the Network

Purpose

Make sure that the expected paths are used.

Action

From operational mode, enter the **traceroute** command on Device CE1.

The first example output shows that a traceroute over the voice topology goes from Device CE1 to Device CE2 where DSCPs are set. The routes are resolved over **:voice.inet.0**. This traceroute path follows the voice path CE1-PE1-P1-P2-PE2-CE2.

`user@CE1>` **traceroute 11.19.140.1 source 11.19.130.1 tos 160**

```
traceroute to 11.19.140.1 (11.19.140.1) from 11.19.130.1, 30 hops max, 40 byte
packets
 1  10.0.0.2 (10.0.0.2)  2.015 ms  1.924 ms  1.770 ms
 2  10.0.0.5 (10.0.0.5)  1.890 ms  1.010 ms  0.974 ms
 3  10.0.0.34 (10.0.0.34)  0.986 ms  1.031 ms  0.973 ms
 4  10.0.0.38 (10.0.0.38)  1.213 ms  1.065 ms  1.154 ms
 5  11.19.140.1 (11.19.140.1)  1.696 ms  4.286 ms  1.332 ms
```

This output shows a traceroute from Device CE1 to Device CE2 for voice where no DSCPs are set. The routes are resolved over **inet.0**, and the resulting path is different from the previous case where the DSCPs are set. This traceroute path follows the default path CE1-PE1-P4-PE2-CE2.

`user@CE1>` **traceroute 11.19.140.1 source 11.19.130.1**

```
traceroute to 11.19.140.1 (11.19.140.1) from 11.19.130.1, 30 hops max, 40 byte
packets
 1  10.0.0.2 (10.0.0.2)  1.654 ms  1.710 ms  1.703 ms
 2  10.0.0.5 (10.0.0.5)  1.790 ms  1.045 ms  0.975 ms
```

```
 3  10.0.0.18 (10.0.0.18)  0.989 ms  1.041 ms  0.983 ms
 4  10.0.0.42 (10.0.0.42)  0.994 ms  1.036 ms  1.002 ms
 5  11.19.140.1 (11.19.140.1)  1.329 ms  2.248 ms  2.225 ms
```

This output shows a traceroute from Device CE1 to Device CE2 for video traffic where the firewall filter is based on the destination address. The routes are resolved over **:video.inet.0**. This traceroute follows the video path CE1-PE1-P3-P4-PE2-CE2.

```
user@CE1> traceroute 11.19.142.1 source 11.19.132.1
```

```
traceroute to 11.19.142.1 (11.19.142.1) from 11.19.132.1, 30 hops max, 40 byte
packets
 1  10.0.0.2 (10.0.0.2)  1.126 ms  1.300 ms  0.995 ms
 2  10.0.0.10 (10.0.0.10)  0.981 ms  1.018 ms  0.991 ms
 3  10.0.0.30 (10.0.0.30)  0.997 ms  1.886 ms  1.952 ms
 4  10.0.0.42 (10.0.0.42)  1.800 ms  1.038 ms  0.980 ms
 5  11.19.142.1 (11.19.142.1)  1.367 ms  1.352 ms  1.328 ms
```

This output shows a traceroute from Device CE1 to Device CE2 for video where DSCPs are set. The DSCP bits are directing Device PE1 to use the topology table **:voice.inet.0**. Because there is no entry in the voice routing table for the video routes, traffic is dropped.

```
user@CE1> traceroute 11.19.142.1 source 11.19.132.1 tos 160
```

```
traceroute to 11.19.142.1 (11.19.142.1) from 11.19.132.1, 30 hops max, 40 byte
packets
 1  10.0.0.2 (10.0.0.2)  1.135 ms !N  1.007 ms !N  0.954 ms !N
```

RELATED DOCUMENTATION

Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | **18**

3

CHAPTER

Monitoring Multitopology Routing

Example: Tracing Global Routing Protocol Operations | **90**

Example: Tracing Global Routing Protocol Operations

IN THIS SECTION

- Requirements | 90
- Overview | 90
- Configuration | 91
- Verification | 95

This example shows how to list and view files that are created when you enable global routing trace operations.

Requirements

You must have the **view** privilege.

Overview

To configure global routing protocol tracing, include the **traceoptions** statement at the **[edit routing-options]** hierarchy level:

```
traceoptions {
   file filename <files number> <size size> <world-readable | no-world-readable>;
   flag flag <disable>;
}
```

The flags in a **traceoptions flag** statement are identifiers. When you use the **set** command to configure a flag, any flags that might already be set are not modified. In the following example, setting the **timer** tracing flag has no effect on the already configured **task** flag. Use the **delete** command to delete a particular flag.

```
[edit routing-options traceoptions]
user@host# show
flag task;
```

```
user@host# set traceoptions flag timer
user@host# show
flag task;
flag timer;
user@host# delete traceoptions flag task
user@host# show
flag timer;
```

This example shows how to configure and view a trace file that tracks changes in the routing table. The steps can be adapted to apply to trace operations for any Junos OS hierarchy level that supports trace operations.

> **TIP:** To view a list of hierarchy levels that support tracing operations, enter the **help apropos traceoptions** command in configuration mode.

Configuration

CLI Quick Configuration

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

```
set routing-options traceoptions file routing-table-changes
set routing-options traceoptions file size 10m
set routing-options traceoptions file files 10
set routing-options traceoptions flag route
set routing-options static route 1.1.1.2/32 next-hop 10.0.45.6
```

Configuring Trace Operations

Step-by-Step Procedure

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

To configure the trace operations:

1. Configure trace operations.

```
[edit routing-options traceoptions]
user@host# set file routing-table-changes
user@host# set file size 10m
user@host# set file files 10
user@host# set flag route
```

2. Configure a static route to cause a change in the routing table.

```
[edit routing-options static]
user@host# set route 1.1.1.2/32 next-hop 10.0.45.6
```

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

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

Viewing the Trace File

Step-by-Step Procedure

To view the trace file:

1. In operational mode, list the log files on the system.

```
user@host> file list /var/log
```

```
/var/log:
...
routing-table-changes
...
```

2. View the contents of the **routing-table-changes** file.

```
user@host> file show /var/log/routing-table-changes
```

```
Dec 15 11:09:29 trace_on: Tracing to "/var/log/routing-table-changes" started
Dec 15 11:09:29.496507
Dec 15 11:09:29.496507 Tracing flags enabled: route
Dec 15 11:09:29.496507
Dec 15 11:09:29.533203 inet_routerid_notify: Router ID: 192.168.4.1
Dec 15 11:09:29.533334 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.533381 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.533420 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.534915 inet_routerid_notify: Router ID: 192.168.4.1
Dec 15 11:09:29.542934 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.549253 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.556878 inet_routerid_notify: No Router ID assigned
Dec 15 11:09:29.582990 rt_static_reinit: examined 3 static nexthops, 0
unreferenced
Dec 15 11:09:29.589920
Dec 15 11:09:29.589920 task_reconfigure reinitializing done
...
```

3. Filter the output of the log file.

   user@host> **file show /var/log/routing-table-changes | match 1.1.1.2**

```
Dec 15 11:15:30.780314 ADD      1.1.1.2/32          nhid 0 gw 10.0.45.6
Static   pref 5/0 metric  at-0/2/0.0 <ctive Int Ext>
Dec 15 11:15:30.782276 KRT Request: send len 216 v104 seq 0 ADD route/user af
2 table 0 infot 0 addr 1.1.1.2 nhop-type unicast nhindex 663
```

4. View the tracing operations in real time by running the **monitor start** command with an optional **match** condition.

   user@host> **monitor start routing-table-changes | match 1.1.1.2**

```
Aug 10 19:21:40.773467 BGP RECV         0.0.0.0/0
Aug 10 19:21:40.773685 bgp_rcv_nlri: 0.0.0.0/0
Aug 10 19:21:40.773778 bgp_rcv_nlri: 0.0.0.0/0 belongs to meshgroup
Aug 10 19:21:40.773832 bgp_rcv_nlri: 0.0.0.0/0 qualified bnp->ribact 0x0 l2afcb
 0x0
```

5. Deactivate the static route.

```
user@host# deactivate routing-options static route 1.1.1.2/32
user@host# commit
```

```
*** routing-table-changes ***
Dec 15 11:42:59.355557 CHANGE   1.1.1.2/32          nhid 663 gw 10.0.45.6
  Static   pref 5/0 metric  at-0/2/0.0 <Delete Int Ext>
Dec 15 11:42:59.426887 KRT Request: send len 216 v104 seq 0 DELETE route/user
af 2 table 0 infot 0 addr 1.1.1.2 nhop-type discard filtidx 0
Dec 15 11:42:59.427366 RELEASE  1.1.1.2/32          nhid 663 gw 10.0.45.6
  Static   pref 5/0 metric  at-0/2/0.0 <Release Delete Int Ext>
```

6. Halt the **monitor** command by pressing Enter and typing **monitor stop**.

```
[Enter]
user@host> monitor stop
```

7. When you are finished troubleshooting, consider deactivating trace logging to avoid any unnecessary impact to system resources.

   When configuration is deactivated, it appears in the configuration with the **inactive** tag.

```
[edit routing-options]
user@host# deactivate traceoptions
user@host# commit
```

```
[edit routing-options]
user@host# show
```

```
inactive: traceoptions {
    file routing-table-changes size 10m files 10;
    flag route;
}
static {
    inactive: route 1.1.1.2/32 next-hop 10.0.45.6;
}
```

8. To reactivate trace operations, use the **activate** configuration-mode statement.

```
[edit routing-options]
user@host# activate traceoptions
user@host# commit
```

Results

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

```
user@host# show routing-options
traceoptions {
    file routing-table-changes size 10m files 10;
    flag route;
}
static {
    route 1.1.1.2/32 next-hop 10.0.45.6;
}
```

Verification

Confirm that the configuration is working properly.

Verifying That the Trace Log File Is Operating

Purpose

Make sure that events are being written to the log file.

Action

`user@host>` **show log routing-table-changes**

```
Dec 15 11:09:29 trace_on: Tracing to "/var/log/routing-table-changes" started
```

4

CHAPTER

Troubleshooting Network Issues

Working with Problems on Your Network | **97**

Isolating a Broken Network Connection | **97**

Identifying the Symptoms of a Broken Network Connection | **99**

Isolating the Causes of a Network Problem | **100**

Taking Appropriate Action for Resolving the Network Problem | **101**

Evaluating the Solution to Check Whether the Network Problem Is Resolved | **103**

Working with Problems on Your Network

Problem

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

Solution

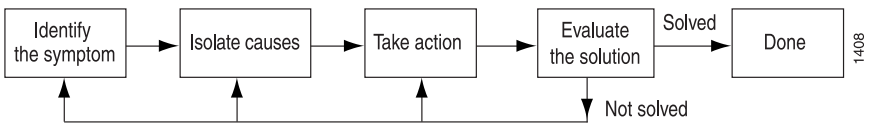
Table 4: Checklist for Working with Problems on Your Network

Tasks	Command or Action
"Isolating a Broken Network Connection" on page 97	
1. Identifying the Symptoms of a Broken Network Connection on page 99	**ping (*ip-address* | *hostname*)** **show route (*ip-address* | *hostname*)** **traceroute (*ip-address* | *hostname*)**
2. Isolating the Causes of a Network Problem on page 100	**show < configuration | interfaces | protocols | route >**
3. Taking Appropriate Action for Resolving the Network Problem on page 101	**[edit]** **delete routing options static route *destination-prefix*** **commit and-quit** **show route *destination-prefix***
4. Evaluating the Solution to Check Whether the Network Problem Is Resolved on page 103	**show route (*ip-address* | *hostname*)** **ping (*ip-address* | *hostname*) count 3** **traceroute (*ip-address* | *hostname*)**

Isolating a Broken Network Connection

By applying the standard four-step process illustrated in Figure 9 on page 97, you can isolate a failed node in the network. Note that the functionality described in this section is not supported in versions 15.1X49, 15.1X49-D30, or 15.1X49-D40.

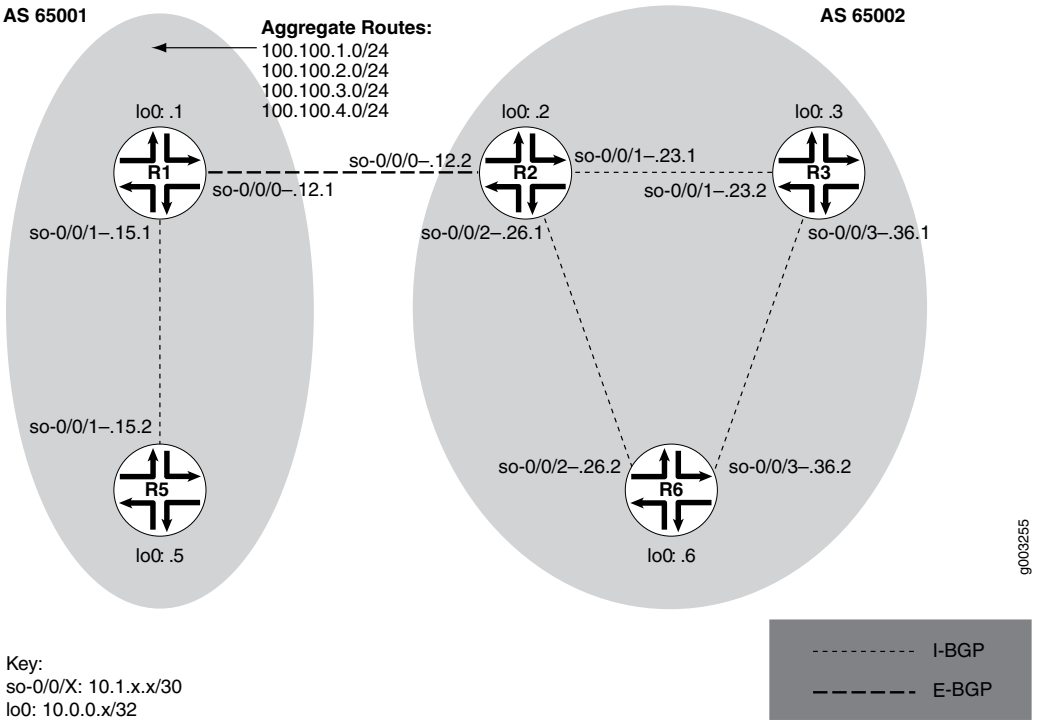
Figure 9: Process for Diagnosing Problems in Your Network



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

Figure 10 on page 98 shows the network topology used in this topic to illustrate the process of diagnosing problems in a network.

Figure 10: Network with a Problem



The network in Figure 10 on page 98 consists of two autonomous systems (ASs). AS 65001 includes two routers, and AS 65002 includes three routers. The border router (**R1**) in AS 65001 announces aggregated prefixes **100.100/24** to the AS 65002 network. The problem in this network is that **R6** does not have access to **R5** because of a loop between **R2** and **R6**.

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

- Isolating the Causes of a Network Problem on page 100
- Taking Appropriate Action for Resolving the Network Problem on page 101

- Taking Appropriate Action for Resolving the Network Problem on page 101
- Evaluating the Solution to Check Whether the Network Problem Is Resolved on page 103

Identifying the Symptoms of a Broken Network Connection

Problem

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

Solution

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

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

Sample Output

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

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

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

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

```
user@R6> show route 10.0.0.5

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

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

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

Meaning

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

Isolating the Causes of a Network Problem

Problem

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

Solution

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

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

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

Sample Output

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

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

```
user@R2> show route 10.0.0.5

inet.0: 22 destinations, 25 routes (22 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

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

Meaning

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

Taking Appropriate Action for Resolving the Network Problem

Problem

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

Solution

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

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

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

Sample Output

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

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

user@R2>  show route 10.0.0.5

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

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

Meaning

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

Evaluating the Solution to Check Whether the Network Problem Is Resolved

Problem

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

You can address possible causes in any order. In relation to the network in "Isolating a Broken Network Connection" on page 97, we chose to work from the local router toward the remote router, but you might start at a different point, particularly if you have reason to believe that the problem is related to a known issue, such as a recent change in configuration.

Solution

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

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

Sample Output

```
user@R6> show route 10.0.0.5

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

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

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

user@R6> traceroute 10.0.0.5
traceroute to 10.0.0.5 (10.0.0.5), 30 hops max, 40 byte packets
```

```
1  10.1.26.1 (10.1.26.1)  0.629 ms  0.538 ms  0.497 ms
2  10.1.12.1 (10.1.12.1)  0.534 ms  0.538 ms  0.510 ms
3  10.0.0.5 (10.0.0.5)  0.776 ms  0.705 ms  0.672 ms
```

Meaning

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

5

CHAPTER

Configuration Statements

rib (Multitopology Routing) | **106**

topologies (Multitopology Routing) | **108**

topology (Filter-Based Forwarding) | **110**

topology (Multitopology Routing) | **112**

topology (OSPF) | **113**

topology (OSPF Interface) | **115**

topology (Protocols BGP) | **117**

topology-id | **119**

rib (Multitopology Routing)

Syntax

```
rib routing-table-name {
    static {
        route destination-prefix {
            next-hop;
        }
        static-options;
    }
}
```

Hierarchy Level

```
[edit logical-systems logical-system-name routing-options],
[edit logical-systems logical-system-name routing-instances routing-instance-name routing-options],
[edit routing-instances routing-instance-name routing-options],
[edit routing-options]
```

Release Information

Statement support for multitopology routing introduced in Junos OS Release 9.0.
Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description

Configure a static route to install routes in the routing table for a specific topology.

Options

routing-table-name—Name of the routing table for a topology. Use the following format: ***logical-system-name/routing-instance-name*:*topology-name*.*protocol*.*identifier***. Include the routing instance string only if the instance is not the master. The logical system string is included only if the logical system identifier has a value other than 0 (zero). Each routing table for a topology includes a colon (:) before the topology name. ***protocol*** is the protocol family, which can be **inet** or **inet6**. ***identifier*** is the positive integer that specifies the instance of the routing table. For example, to install IPv6 routes to the routing table for a topology named voice in the master instance, include **:voice.inet6.0**.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

- *Example: Removing the Class E Prefix on Martian Addresses*
- *Example: Configuring IPv6 BGP Routes over IPv4 Transport*
- *Enabling Layer 2 VPN and VPLS Signaling*
- *Understanding Martian Addresses*
- *static*

topologies (Multitopology Routing)

Syntax

```
topologies {
    family (inet | inet6) {
        topology topology-name;
    }
}
```

Hierarchy Level

```
[edit logical-systems logical-system-name routing-options],
[edit logical-systems logical-system-name routing-instances routing-instance-name routing-options],
[edit routing-instances routing-instance-name routing-options],
[edit routing-options]
```

Release Information

Statement introduced in Junos OS Release 9.0.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

Description

Configure a topology for multitopology routing. Each topology creates a new routing table that is populated with direct routes from the topology.

Options

family—Configure the type of family address type.

inet—IPv4

inet6—IPv6

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths | 55

Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 20

Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 18

Understanding Multitopology Routing in Conjunction with PIM | 52

topology (Filter-Based Forwarding)

Syntax

topology *topology-name*;

Hierarchy Level

[edit firewall family (inet | inet6) filter *filter-name* term *term-name* then],
[edit firewall family (inet | inet6) filter *filter-name* term *term-name* then logical-system *logical-system-name*],
[edit firewall family (inet | inet6) filter *filter-name* term *term-name* then logical-system *logical-system-name* routing-instance *routing-instance-name*],
[edit firewall family (inet | inet6) filter *filter-name* term *term-name* then routing-instance *routing-instance-name*]

Release Information

Statement introduced in Junos OS Release 9.0.

Description

Configure a topology for filter-based forwarding for multitopology routing. The firewall filter you apply to the ingress interface is used to look up traffic against the configured topology, and, if a route matches the conditions you configure for the term, the route is accepted and added to the routing table for the specific topology.

There are multiple ways to configure a topology for filter-based forwarding, depending on the type of instance or logical system you want to specify for the forwarding class.

NOTE: The options for logical system and routing instance precede the **topology** statement with the **then** statement.

Options

topology-name—Name of a topology against which you want to match traffic.

Required Privilege Level

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

RELATED DOCUMENTATION

Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 20

Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 18

topology (Multitopology Routing)

Syntax

```
topology topology-name;
```

Hierarchy Level

```
[edit logical-systems logical-system-name routing-instances routing-instance-name routing-options topologies family
    (inet | inet6)],
[edit logical-systems logical-system-name routing-options topologies family (inet | inet6)],
[edit routing-instances routing-instance-name routing-options topologies family (inet | inet6)],
[edit routing-options topologies family (inet | inet6)]
```

Release Information

Statement introduced in Junos OS Release 9.0.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

Description

Configure the name of a topology configured to run multitopology routing.

Options

topology-name—Name of the topology. Include a string value that describes the type of traffic, such as voice or video. For IPv4 multicast traffic, include **ipv4-multicast** as the name.

Required Privilege Level

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

RELATED DOCUMENTATION

- Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 20
- Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths | 55
- Understanding Multitopology Routing in Conjunction with PIM | 52
- Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 18

topology (OSPF)

Syntax

```
topology (default | ipv4-multicast | name) {
    spf-options {
        delay milliseconds;
        holddown milliseconds;
        rapid-runs number;
    }
    topology-id number;
}
```

Hierarchy Level

```
[edit logical-systems logical-system-name protocols ospf],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols ospf],
[edit protocols ospf],
[edit routing-instances routing-instance-name protocols ospf]
```

Release Information

Statement introduced in Junos OS Release 9.0.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Enable a topology for OSPF multitopology routing. You must first configure one or more topologies under the **[edit routing-options]** hierarchy level.

Options

default—Name of the default topology. This topology is automatically created, and all routes that correspond to it are automatically added to the **inet.0** routing table. You can modify certain default parameters, such as for the SPF algorithm.

ipv4-multicast—Name of the topology for IPv4 multicast traffic.

name—Name of a topology you configured at the **[edit routing-options]** hierarchy level to create a topology for a specific type of traffic, such as voice or video.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

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

RELATED DOCUMENTATION

Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths | 55

Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 20

Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 18

Understanding Multitopology Routing in Conjunction with PIM | 52

topology (OSPF Interface)

Syntax

```
topology (ipv4-multicast | name) {
    metric metric;
}
```

Hierarchy Level

```
[edit logical-systems logical-system-name protocols ospf area area-id interface interface-name],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols ospf area area-id interface
    interface-name],
[edit protocols ospf area area-id interface interface-name],
[edit routing-instances routing-instance-name protocols ospf area area-id interface interface-name]
```

Release Information

Statement introduced in Junos OS Release 9.0.

Description

Configure interface-specific properties for multitopology OSPF, including topology-specific metric values for an interface.

All OSPF interfaces have a cost, which is a routing metric that is used in the link-state calculation. Routes with lower total path metrics are preferred over those with higher path metrics. The default value for the OSPF metric for an interface is 1. You can modify the default value for an OSPF interface and configure a topology-specific metric for that interface. The topology-specific metric applies to routes advertised from the interface that belong only to that topology.

Default

The default value of the topology metric is the same as the default metric value calculated by OSPF or the value configured for the OSPF metric.

Options

ipv4-multicast—Name of the topology for IPv4 multicast traffic.

name—Name of a topology created under the **[edit routing-options]** hierarchy level.

metric *metric*—Cost of a route from an OSPF interface. You can specify a metric value for a topology that is different from the value specified for the interface.

Range: 1 through 65,535

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: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | **20**
- Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths | **55**
- Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | **18**
- Understanding Multitopology Routing in Conjunction with PIM | **52**

topology (Protocols BGP)

Syntax

```
topology name {
    community {
        target identifier;
    }
}
```

Hierarchy Level

```
[edit logical-systems logical-system-name protocols bgp family (inet | inet6) unicast],
[edit logical-systems logical-system-name protocols bgp group group-name family (inet | inet6) unicast],
[edit logical-systems logical-system-name protocols bgp group group-name neighbor address family (inet | inet6)
    unicast],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols bgp family (inet | inet6)
    unicast],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols bgp group group-name
    family (inet | inet6) unicast],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols bgp group group-name
    neighbor address family (inet | inet6) unicast],
[edit protocols bgp family (inet | inet6) unicast],
[edit protocols bgp group group-name family (inet | inet6) unicast],
[edit protocols bgp group group-name neighbor address family (inet | inet6) unicast],
[edit routing-instances routing-instance-name protocols bgp family (inet | inet6) unicast],
[edit routing-instances routing-instance-name protocols bgp group group-name family (inet | inet6) unicast],
[edit routing-instances routing-instance-name protocols bgp group group-name neighbor address family (inet | inet6)]
```

Release Information

Statement introduced in Junos OS Release 9.0.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Enable a topology for BGP multitopology routing. You must first configure one or more topologies under the **[edit routing-options]** hierarchy level.

Apply the community tags to identify the application topologies by configuring a routing topology name and BGP community value.

In Junos OS, multitopology support for BGP is based on the community value in a BGP route. This configuration determines the association between a topology and one or more community values and populates the topology routing tables. Arriving BGP updates that have a matching community value are

replicated in the associated topology routing table. You decide which BGP community values are associated with a given topology.

For example, you can create a configuration that causes BGP updates that are received with community value **target:40:40** to be added into topology routing table **voice.inet.0** (in addition to the default routing table **inet.0**). Likewise, you configuration can specify that updates that are received with community value **target:50:50** are added into topology routing table **video.inet.0** (in addition to the default routing table **inet.0**).

Options

name—Name of a topology you configured at the **[edit routing-options]** hierarchy level to create a topology for a specific type of traffic, such as voice or video.

community—Configure the community to identify the multitopology routes. BGP uses the target community identifier to install the routes it learns in the appropriate multitopology routing tables.

Syntax: target *identifier*—Configure the destination to which the route is going.

Required Privilege Level

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

RELATED DOCUMENTATION

- Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths | **55**
- Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | **20**
- Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | **18**
- Understanding Multitopology Routing in Conjunction with PIM | **52**

topology-id

Syntax

```
topology-id number;
```

Hierarchy Level

```
[edit logical-systems logical-system-name protocols ospf topology name],
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols ospf topology name],
[edit protocols ospf topology name],
[edit routing-instances routing-instance-name protocols ospf topology name]
```

Release Information

Statement introduced in Junos OS Release 9.0.

Description

Configure a topology identifier for a topology enabled for OSPF.

Default

The default identifier for the default topology is 0, and the default identifier for the topology for IPv4 multicast traffic is 1. These identifiers are predefined and cannot be modified.

Options

number—The integer value used to identify the topology.

Range: 32 through 127

Required Privilege Level

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

RELATED DOCUMENTATION

Example: Configuring Multitopology Routing to Provide Redundancy for Multicast Traffic over Separate Network Paths | 55

Example: Configuring Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 20

Understanding Multitopology Routing for Class-Based Forwarding of Voice, Video, and Data Traffic | 18

Understanding Multitopology Routing in Conjunction with PIM | 52

topology | 113

CHAPTER 6

Operational Commands

show (ospf | ospf3) interface | **121**

show (ospf | ospf3) interface

List of Syntax

Syntax on page 121
Syntax (EX Series Switches and QFX Series) on page 121

Syntax

```
show (ospf | ospf3) interface
<brief | detail | extensive>
<area area-id>
<interface-name>
<instance instance-name>
<logical-system (all | logical-system-name)>
<realm (ip4-multicast | ipv4-unicast | ipv6-multicast)>
```

Syntax (EX Series Switches and QFX Series)

```
show (ospf | ospf3) interface
<brief | detail | extensive>
<area area-id>
<interface-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.
area option introduced in Junos OS Release 9.2.
area option introduced in Junos OS Release 9.2 for EX Series switches.
realm option introduced in Junos OS Release 9.2.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display the status of OSPF interfaces.

Options

none—Display standard information about the status of all OSPF interfaces for all routing instances

brief | detail | extensive—(Optional) Display the specified level of output.

area *area-id*—(Optional) Display information about the interfaces that belong to the specified area.

interface-name—(Optional) Display information for the specified interface.

instance *instance-name*—(Optional) Display all OSPF interfaces under the named routing instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

realm (ipv4-multicast | ipv4-unicast | ipv6-multicast)—(OSPFv3 only) (Optional) Display information about the interfaces for the specified OSPFv3 realm, or address family. Use the **realm** option to specify an address family for OSPFv3 other than IPv6 unicast, which is the default.

Required Privilege Level
view

List of Sample Output
show ospf interface brief on page 125
show ospf interface detail on page 125
show ospf3 interface detail on page 126

Output Fields
Table 5 on page 122 lists the output fields for the **show (ospf | ospf3) interface** command. Output fields are listed in the approximate order in which they appear.

Table 5: show (ospf | ospf3) interface Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the interface running OSPF version 2 or OSPF version 3.	All levels
State	State of the interface: BDR, Down, DR, DRother, Loop, PtToPt, or Waiting.	All levels
Area	Number of the area that the interface is in.	All levels
DR ID	Address of the area's designated router.	All levels
BDR ID	Backup designated router for a particular subnet.	All levels
Nbrs	Number of neighbors on this interface.	All levels
Type	Type of interface: LAN, NBMA, P2MP, P2P, or Virtual.	**detail extensive**
Address	IP address of the neighbor.	**detail extensive**
Mask	Netmask of the neighbor.	**detail extensive**
Prefix-length	(OSPFv3) IPv6 prefix length, in bits.	**detail extensive**
OSPF3-Intf-Index	(OSPFv3) OSPF version 3 interface index.	**detail extensive**

Table 5: show (ospf | ospf3) interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
MTU	Interface maximum transmission unit (MTU).	**detail extensive**
Cost	Interface cost (metric).	**detail extensive**
DR addr	Address of the designated router.	**detail extensive**
BDR addr	Address of the backup designated router.	**detail extensive**
Adj count	Number of adjacent neighbors.	**detail extensive**
Secondary	Indicates that this interface is configured as a secondary interface for this area. This interface can belong to more than one area, but can be designated as a primary interface for only one area.	**detail extensive**
Flood Reduction	Indicates that this interface is configured with flooding reduction. All self-originated LSAs from this interface are initially sent with the **DoNotAge** bit set. As a result, LSAs are refreshed only when a change occurs.	**extensive**
Priority	Router priority used in designated router (DR) election on this interface.	**detail extensive**
Flood list	List of link-state advertisements (LSAs) that might be about to flood this interface.	**extensive**
Ack list	Acknowledgment list. List of pending acknowledgments on this interface.	**extensive**
Descriptor list	List of packet descriptors.	**extensive**
Hello	Configured value for the hello timer.	**detail extensive**
Dead	Configured value for the dead timer.	**detail extensive**
Auth type	(OSPFv2) Authentication mechanism for sending and receiving OSPF protocol packets: • MD5—The MD5 mechanism is configured in accordance with RFC 2328. • None—No authentication method is configured. • Password—A simple password (RFC 2328) is configured.	**detail extensive**
Topology	(Multiarea adjacency) Name of topology: default or *name*.	**detail extensive**

Table 5: show (ospf | ospf3) interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
LDP sync state	(OSPFv2 and LDP synchronization) Current state of LDP synchronization: in sync, in holddown, and not supported.	**extensive**
reason	(OSPFv2 and LDP synchronization) Reason for the current state of LDP synchronization. The LDP session might be up or down, or adjacency might be up or down.	**extensive**
config holdtime	(OSPFv2 and LDP synchronization) Configured value of the hold timer. If the state is not synchronized, and the hold time is not infinity, the **remaining** field displays the number of seconds that remain until the configured hold timer expires.	**extensive**
IPSec SA name	(OSPFv2) Name of the IPSec security association name.	**detail extensive**
Active key ID	(OSPFv2 and MD5) Number from 0 to 255 that uniquely identifies an MD5 key.	**detail extensive**
Start time	(OSPFv2 and MD5) Time at which the routing device starts using an MD5 key to authenticate OSPF packets transmitted on the interface on which this key is configured. To authenticate received OSPF protocol packets, the key becomes effective immediately after the configuration is committed. If the start time option is not configured, the key is effective immediately for send and receive and is displayed as Start time 1970 Jan 01 00:00:00 PST.	**detail extensive**
ReXmit	Configured value for the Retransmit timer.	**detail extensive**
Stub, Not Stub, or Stub NSSA	Type of area.	**detail extensive**

Table 5: show (ospf | ospf3) interface Output Fields ***(continued)***

Field Name	Field Description	Level of Output
Post convergence Protection	Post convergence protection can have the following types when enabled • **Fate Sharing** can have the following values • **Yes**-You have configured fate-sharing protection. • **No**-You have not configured fate-sharing protection. • **node protection** can have the following values: • **Yes**-You have configured node protection. • **No**-You have not configured node protection. • **srlg protection** can have the following values: • **Yes**–You have configured Shared Risk Link Groutp (SRLG) protection. • **No**–You have not configured SRLG protection. **Node cost** is the recalculated metric cost of the node.	**extensive**

Sample Output

show ospf interface brief

```
user@host> show ospf interface brief
```

```
Intf                State     Area            DR ID           BDR ID       Nbrs
at-5/1/0.0          PtToPt    0.0.0.0         0.0.0.0         0.0.0.0        1
ge-2/3/0.0          DR        0.0.0.0         192.168.4.16    192.168.4.15   1
lo0.0               DR        0.0.0.0         192.168.4.16    0.0.0.0        0
so-0/0/0.0          Down      0.0.0.0         0.0.0.0         0.0.0.0        0
so-6/0/1.0          PtToPt    0.0.0.0         0.0.0.0         0.0.0.0        1
so-6/0/2.0          Down      0.0.0.0         0.0.0.0         0.0.0.0        0
so-6/0/3.0          PtToPt    0.0.0.0         0.0.0.0         0.0.0.0        1
```

show ospf interface detail

```
user@host> show ospf interface detail
```

```
Interface             State     Area            DR ID           BDR ID Nbrs
fe-0/0/1.0            BDR      0.0.0.0         192.168.37.12   10.255.245.215 1
Type LAN, address 192.168.37.11, Mask 255.255.255.248, MTU 4460, Cost 40
DR addr 192.168.37.12, BDR addr 192.168.37.11, Adj count 1, Priority 128
Hello 10, Dead 40, ReXmit 5, Not Stub
```

```
t1-0/2/1.0             PtToPt   0.0.0.0         0.0.0.0         0.0.0.0 0
Type P2P, Address 0.0.0.0, Mask 0.0.0.0, MTU 1500, Cost 2604
  Adj count 0
Hello 10, Dead 40, ReXmit 5, Not Stub
Auth type: MD5, Active key ID 3, Start time 2002 Nov 19 10:00:00 PST
IPsec SA Name: sa
```

show ospf3 interface detail

user@host> **show ospf3 interface so-0/0/3.0 detail**

```
Interface              State     Area           DR-ID         BDR-ID     Nbrs
so-0/0/3.0             PtToPt   0.0.0.0        0.0.0.0       0.0.0.0       1
Address fe80::2a0:a5ff:fe28:1dfc, Prefix-length 64
OSPF3-Intf-index 1, Type P2P, MTU 4470, Cost 12, Adj-count 1
Hello 10, Dead 40, ReXmit 5, Not Stub
```

show ospf interface extensive (SRLG Protection Enabled)

user@host> **show ospf interface extensive**

```
Interface           State   Area            DR ID           BDR ID          Nbrs
ge-0/0/0.0          DR      0.0.0.0         10.205.172.20   10.205.171.195     1
  Type: LAN, Address: 81.1.2.1, Mask: 255.255.255.0, MTU: 1500, Cost: 1
  DR addr: 81.1.2.1, BDR addr: 81.1.2.2, Priority: 128
  Adj count: 1
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
  Protection type: Post Convergence
  Post convergence protection: Enabled, Fate sharing: No, SRLG: Yes, Node cost:
65535
  Topology default (ID 0) -> Cost: 1
• Checking backup route in rib:
root@R0# run show route 6.6.6.6
inet.0: 61 destinations, 61 routes (61 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

6.6.6.6/32         *[OSPF/10] 00:08:52, metric 20
                    > to 41.41.41.2 via ge-0/0/1.0
                    > to 31.31.31.2 via ge-0/0/2.0, Push 800030
```

```
inet.3: 6 destinations, 10 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

6.6.6.6/32         *[LDP/9] 00:07:33, metric 1
                    > to 41.41.41.2 via ge-0/0/1.0, Push 299808
                    [L-OSPF/10/5] 00:07:33, metric 20
                    > to 41.41.41.2 via ge-0/0/1.0, Push 800060
                    > to 31.31.31.2 via ge-0/0/2.0, Push 800030, Push 800060
```

`user@host>` **show ospf interface extensive (Fate-Sharing Protection Enabled)**

```
Interface           State   Area            DR ID           BDR ID          Nbrs
ge-0/0/0.0          DR      0.0.0.0         10.205.172.20   10.205.171.195     1
  Type: LAN, Address: 81.1.2.1, Mask: 255.255.255.0, MTU: 1500, Cost: 1
  DR addr: 81.1.2.1, BDR addr: 81.1.2.2, Priority: 128
  Adj count: 1
  Hello: 10, Dead: 40, ReXmit: 5, Not Stub
  Auth type: None
  Protection type: Post Convergence
  Post convergence protection: Enabled, Fate sharing: Yes, SRLG: No, Node cost:
65535
  Topology default (ID 0) -> Cost: 1
• Checking backup route in rib:
root@R0# run show route 6.6.6.6
inet.0: 61 destinations, 61 routes (61 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

6.6.6.6/32         *[OSPF/10] 00:08:52, metric 20
                    > to 41.41.41.2 via ge-0/0/1.0
                    > to 31.31.31.2 via ge-0/0/2.0, Push 800030

inet.3: 6 destinations, 10 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

6.6.6.6/32         *[LDP/9] 00:07:33, metric 1
                    > to 41.41.41.2 via ge-0/0/1.0, Push 299808
                    [L-OSPF/10/5] 00:07:33, metric 20
                    > to 41.41.41.2 via ge-0/0/1.0, Push 800060
                    > to 31.31.31.2 via ge-0/0/2.0, Push 800030, Push 800060
```