JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

ICMP Router Discovery Protocol User Guide

Published
2024-12-17

JUNOS

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

Junos® OS ICMP Router Discovery Protocol 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 This Guide

Use this guide to configure, monitor and troubleshoot the ICMP Router Discovery protocol on Juniper Networks devices.

1
CHAPTER

Overview

ICMP Router Discovery Overview

IN THIS SECTION

Router discovery uses Internet Control Message Protocol (ICMP) router advertisements and router solicitation messages to allow a host to discover the addresses of operational routers on the subnet. Hosts must discover routers before they can send IP datagrams outside their subnet. Router discovery allows a host to discover the addresses of operational routers on the subnet.

Router discovery uses Internet Control Message Protocol (ICMP) router advertisements and router solicitation messages to allow a host to discover the addresses of operational routers on the subnet. Hosts must discover routers before they can send IP datagrams outside their subnet.

Router discovery allows a host to discover the addresses of operational routers on the subnet. The Junos® operating system (Junos OS) implementation of router discovery supports server mode only.

Each router periodically multicasts a router advertisement from each of its multicast interfaces, announcing the IP address of that interface. Hosts listen for advertisements to discover the addresses of their neighboring routers. When a host starts, it can send a multicast router solicitation to ask for immediate advertisements.

The router discovery messages do not constitute a routing protocol. They enable hosts to discover the existence of neighboring routers, but do not determine which router is best to reach a particular destination.

This section discusses the following topics:

Operation of a Router Discovery Server

The router discovery server distributes information about the addresses of all routers on directly connected networks and about their preferences for becoming the default router. (A host sends a packet to the default router if the host does not have a route to a destination in its routing table.) The server does this by periodically sending router advertisement packets out each interface on which router

discovery is enabled. In addition to containing the router addresses, these packets also announce the existence of the server itself.

The server can either transmit broadcast or multicast router advertisement packets. Multicast packets are sent to 224.0.0.1, which is the all-hosts multicast address. When packets are sent to the all-hosts multicast address, or when an interface is configured for the limited-broadcast address 255.255.255.255, all IP addresses configured on the physical interface are included in the router advertisement. When the packets are being sent to a network or subnet broadcast address, only the address associated with that network or subnet is included in the router advertisement.

When the routing protocol process first starts on the server router, the server sends router advertisement packets every few seconds. Then, the server sends these packets less frequently, commonly every 10 minutes.

The server responds to router solicitation packets it receives from a client. The response is sent unicast unless a router advertisement packet is due to be sent out momentarily.

NOTE: Junos OS does not support the ICMP router solicitation message with the source address as **0.0.0.0**.

Router Advertisement Messages

Router advertisement messages include a preference level and a lifetime field for each advertised router address.

The preference level specifies the router's preference to become the default router. When a host chooses a default router address, it chooses the address with the highest preference. You can configure the preference level by including the `priority` statement.

The lifetime field indicates the maximum length of time that the advertised addresses are to be considered valid by hosts in the absence of further advertisements. You can configure the advertising rate by including the **max-advertisement-interval** and `min-advertisement-interval` statements, and you can configure the lifetime by including the `lifetime` statement. .

RELATED DOCUMENTATION

Example: Configuring the ICMP Protocol for Discovering Gateways to Other Networks | 7

Supported ICMP Router Discovery and IPv6 Neighbor Discovery Standards

Junos OS substantially supports the following RFCs, which define standards for the Internet Control Message Protocol (ICMP for IP version 4 [IPv4]) and neighbor discovery (for IP version 6 [IPv6]).

- RFC 1256, *ICMP Router Discovery Messages*
- RFC 4861, *Neighbor Discovery for IP version 6 (IPv6)*
- RFC 2463, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
- RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
-
- RFC 4862, *IPv6 Stateless Address Autoconfiguration*
- RFC 8335, *PROBE: A Utility for Probing Interfaces*

CHAPTER 2

Configuring the ICMP Protocol

Understanding the ICMP Protocol for Discovering Gateways to Other Networks

The ICMP Router Discovery Protocol (IRDP) enables hosts to locate routers on the local subnet and use them as a gateway to reach other networks. Junos OS supports running IRDP in server mode, meaning that router discovery packets are generated. Junos OS does not support IRDP in client mode running as a host sending router solicitation messages. IRDP is specified in RFC 1256, *ICMP Router Discovery Messages*.

For a host to participate on an internetwork, it needs connectivity to at least one router on the local network. One way to ensure that this is the case is to manually configure each host with the address of a local router as its default router (also called a *gateway*). This method is time-consuming to set up, difficult to maintain, and inflexible.

When you enable the Dynamic Host Configuration Protocol (DHCP) on a host, you do not need to configure the default router. DHCP uses a method called router discovery to automatically discover local routers, and learn other information about them.

The information provided includes the router's address (or addresses, if it has more than one) and how long the host should retain information about the router. Router advertisement messages are sent periodically. Hosts listen for these messages. When an advertisement is received, the host processes it and adds the information about the router to its routing table. A host that has no manually configured routing information has no connectivity to routers when it first powers on. Instead of waiting for the next Router Advertisement message, the host sends a router solicitation message on its local network. This prompts any router that receives this message to immediately send an extra router advertisement message directly to that host.

By default, router discovery is disabled on Junos OS routing devices. When router discovery is enabled, the default behavior is to advertise all interfaces. If the router supports multicast, all the IPv4 Layer 3 interfaces are advertised through multicast. Otherwise, all the IPv4 Layer 3 interfaces are advertised through broadcast.

RELATED DOCUMENTATION

Example: Configuring the ICMP Protocol for Discovering Gateways to Other Networks | 7

Example: Configuring the ICMP Protocol for Discovering Gateways to Other Networks

IN THIS SECTION

This example shows how to configure Internet Control Message Protocol (ICMP) router advertisements to allow IPv4 hosts to discover the addresses of operational routers on the subnet. Hosts must discover routers before they can send IP datagrams outside their subnet.

Requirements

This example assumes that a server or a client computer on the local network supports RFC 1256, *ICMP Router Discovery Messages*.

Overview

IN THIS SECTION

Before a host is able to send a message to a host outside its own subnet, it must be able to identify the address of the immediate router. This is typically done through reading a configuration file upon startup, and on some multicast networks by listening to routing protocol traffic. When a server or a client

computer on the local network that supports RFC 1256 needs to locate a default gateway (router), the server or client computer uses ICMP to send a router solicitation. Hosts that support RFC 1256 send an ICMP router discovery message on the multicast address 224.0.0.2. Routers on the local network that support RFC 1256 immediately respond with a router advertisement.

The all-routers IP multicast address, 224.0.0.2, is the local IP broadcast address that IPv4 reserved. IPv4 multicast addresses in the range 224.0.0.0/24 (from 224.0.0.0 to 224.0.0.255) are reserved for the local subnet.

The ICMP Router Discovery Protocol (IRDP) uses router advertisements as well as router solicitation messages to allow hosts to learn the IP addresses of the router that is attached to the immediate network. When a host is started, it sends router solicitation messages to check for the address of the immediate router.

NOTE: Not all hosts perform router discovery using the method specified in RFC 1256. If the host has DHCP enabled, it might not use ICMP router discovery. The performance of router discovery is one of the DHCP options that is defined in RFC 1541, *Dynamic Host Configuration Protocol*. This option specifies whether the client solicits routers using the ICMP router discovery method specified in RFC 1256. A value of 1 indicates that the client performs router discovery. A value of 0 indicates that the client does not.

To configure the router to be a router discovery server, you must include at least the following statement in the configuration. All other router discovery configuration statements are optional.

```
[edit]
protocols {
    router-discovery;
}
```

To configure a router as a server for ICMP router discovery, you can include the following statements in the configuration:

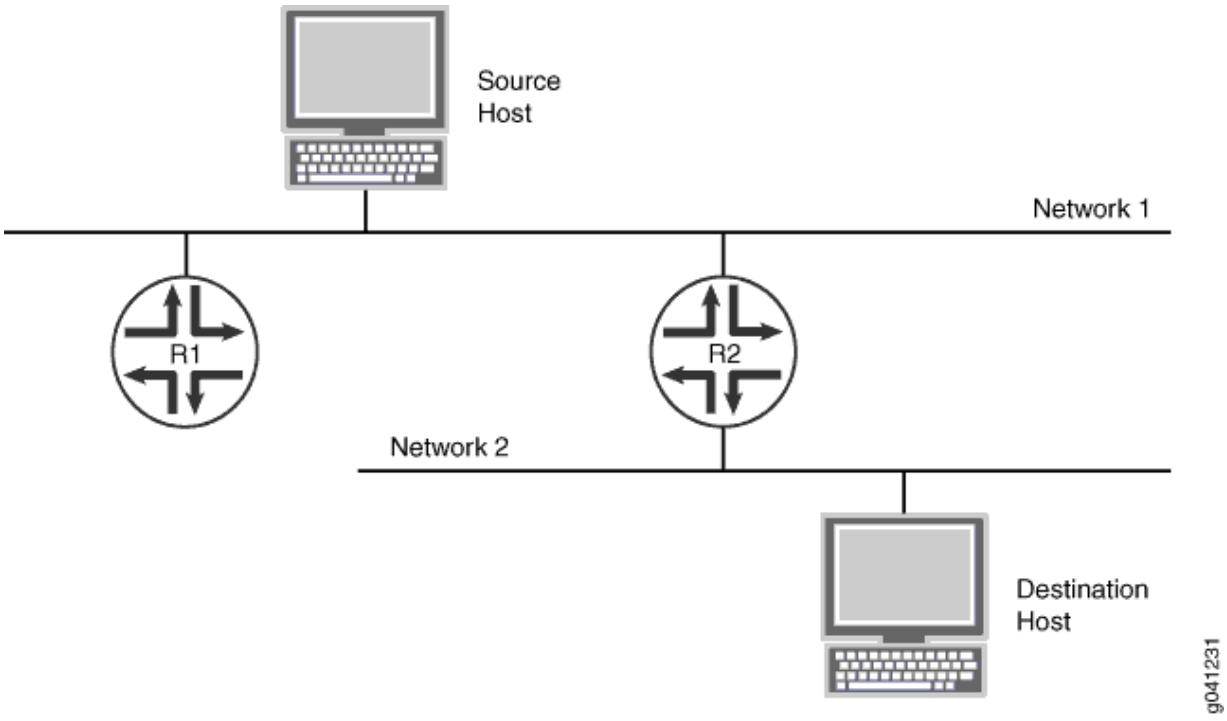
```
[edit]
protocols {
    router-discovery {
        disable;
        address address {
            (advertise | ignore);
            (broadcast | multicast);
```

```
            (ineligible | priority number);
        }
        interface interface-name {
            lifetime seconds;
            max-advertisement-interval seconds;
            min-advertisement-interval seconds;
        }
        traceoptions {
            file filename <files number> <size size> <world-readable | no-world-readable>;
            flag flag <detail> <disable>;
        }
    }
}
```

Topology

Figure 1 on page 9 shows a simplified sample topology.

Figure 1: ICMP Router Discovery Topology



Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

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

```
set interfaces ge-1/2/0 unit 6 description to-R2
set interfaces ge-1/2/0 unit 6 family inet address 10.0.0.6/24
set protocols router-discovery traceoptions file icmp-log
set protocols router-discovery traceoptions flag all
set protocols router-discovery interface ge-1/2/0.6 max-advertisement-interval 60
set protocols router-discovery interface ge-1/2/0.6 min-advertisement-interval 10
set protocols router-discovery interface ge-1/2/0.6 lifetime 120
set protocols router-discovery address 10.0.0.6 multicast
set protocols router-discovery address 10.0.0.6 priority 900
```

Step-by-Step Procedure

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

To configure ICMP router discovery:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```
[edit interfaces]
user@R1# set ge-1/2/0 unit 6 description to-R2
user@R1# set ge-1/2/0 unit 6 family inet address 10.0.0.6/24
```

2. Enable router discovery.

```
[edit protocols]
user@R1# set router-discovery
```

3. (Optional) Enable trace operations for router discovery.

```
[edit protocols router-discovery]
user@R1# set traceoptions file icmp-log
user@R1# set traceoptions flag all
```

4. (Optional) Set the IRDP maximum interval between advertisements.

```
[edit protocols router-discovery]
user@R1# set interface ge-1/2/0.6 max-advertisement-interval 60
```

5. (Optional) Set the IRDP minimum interval between advertisements.

```
[edit protocols router-discovery]
user@R1# set interface ge-1/2/0.6 min-advertisement-interval 10
```

6. (Optional) Set the IRDP period for which advertisements are valid.

```
[edit protocols router-discovery]
user@R1# set interface ge-1/2/0.6 lifetime 120
```

7. (Optional) Configure the router to include the 10.0.0.6 IP address in IRDP advertisements to the all-hosts multicast address (224.0.0.1).

If the router supports IP multicast, and if the interface supports IP multicast, **multicast** is the default. Otherwise, the addresses are included in broadcast router advertisement packets.

```
[edit protocols router-discovery]
user@R1# set address 10.0.0.6 multicast
```

8. (Optional) Set the preference of the address to become a default router.

   This preference is set relative to the preferences of other router addresses on the same subnet.

```
[edit protocols router-discovery]
user@R1# set address 10.0.0.6 priority 900
```

Results

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

```
user@R1# show interfaces
ge-1/2/0 {
    unit 6 {
        description to-R2;
        family inet {
            address 10.0.0.6/24;
        }
    }
}
```

```
user@R1# show protocols
router-discovery {
    traceoptions {
        file icmp-log;
        flag all;
    }
    interface ge-1/2/0.6 {
        max-advertisement-interval 60;
        min-advertisement-interval 10;
        lifetime 120;
```

```
    }
    address 10.0.0.6 {
        multicast;
        priority 900;
    }
}
```

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

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Checking the Trace Log

Purpose

Verify that the expected interfaces are sending messages.

Action

From operational mode, enter the `show log icmp-log` command.

```
user@R1> show log icmp-log
Mar 21 14:42:54 trace_on: Tracing to "/var/log/icmp-log" started
Mar 21 14:42:54.409027 rdisc_ifa_change: Preference for address 10.0.0.6(ge-1/2/0.6) set to 900
Mar 21 14:43:33.983695 task_timer_uset: timer RouterDiscoveryServer_Group <Touched Processing>
set to offset 22 at 14:43:16
Mar 21 14:43:33.984263 rdisc_server_timer: group ge-1/2/0.6 timer set to 22
Mar 21 14:43:55.985225 task_timer_uset: timer RouterDiscoveryServer_Group <Touched Processing>
set to offset 37 at 14:44:10
Mar 21 14:43:55.985520 rdisc_server_timer: group ge-1/2/0.6 timer set to 37
```

```
Mar 21 14:44:32.986407 task_timer_uset: timer RouterDiscoveryServer_Group <Touched Processing>
set to offset 39 at 14:44:44
Mar 21 14:44:32.986961 rdisc_server_timer: group ge-1/2/0.6 timer set to 39
Mar 21 14:45:11.987331 task_timer_uset: timer RouterDiscoveryServer_Group <Touched Processing>
set to offset 10 at 14:44:42
Mar 21 14:45:11.987888 rdisc_server_timer: group ge-1/2/0.6 timer set to 10
Mar 21 14:45:21.990974 task_timer_uset: timer RouterDiscoveryServer_Group <Touched Processing>
set to offset 23 at 14:45:34
Mar 21 14:45:21.991548 rdisc_server_timer: group ge-1/2/0.6 timer set to 23
Mar 21 14:45:44.992150 task_timer_uset: timer RouterDiscoveryServer_Group <Touched Processing>
set to offset 45 at 14:46:06
Mar 21 14:45:44.992710 rdisc_server_timer: group ge-1/2/0.6 timer set to 45
```

Meaning

The log output shows that the preference was set to 900 for IP address 10.0.0.6 and that messages are being sent on the ge-1/2/0.6 interface.

RELATED DOCUMENTATION

ICMP Router Discovery Overview | 2

Understanding the ICMP Protocol for Discovering Gateways to Other Networks | 6

CHAPTER 3

Configuring Recursive DNS Servers

Understanding Recursive DNS Servers for IPv6

To access any location on the Internet, the domain name system (DNS) server plays a pivotal role in resolving the domain name into its associated IP address. The DNS resolution service can also be provided by the DHCP server. The routing protocol process (rpd) of routers generates router advertisements to facilitate IPv6 hosts in autoconfiguration and in learning network information. For IPv6 stateless autoconfiguration, DNS configuration is provided by router advertisements. The router advertisement-based DNS configuration is useful in networks where an IPv6 host's address is autoconfigured through an IPv6 stateless address and where there is no existing DHCPv6 infrastructure.

Depending on their configuration, DNS servers can be classified into the following types:

- Recursive domain name system
- Nonrecursive domain name system

DNS servers can resolve either recursive or nonrecursive queries. For a recursive query by a DNS client, the DNS server returns either the IP address associated with the domain name or an error. A recursive query does not return a referral. For a nonrecursive query, the DNS server returns the IP address of the domain name or an error or a referral to another DNS server which might have the resolution of the query.

For IPv6 hosts, a maximum of three recursive DNS server addresses can be configured along with their respective lifetimes. The default value of the lifetime of the configured recursive DNS server addresses is 1800 seconds. The configured IPv6 host uses the specified recursive DNS server address for DNS resolution where the IPv6 host's address is autoconfigured through an IPv6 stateless address and where there is no DHCPv6 infrastructure available.

CAUTION: The recursive DNS server configuration is included in the router advertisement packet, which is a part of the Neighbor Discovery Protocol (NDP). In general, in an unsecured deployment scenario, an attacker could send a router advertisement with a fraudulent recursive DNS server address, misleading the IPv6 host into contacting an unintended DNS server for DNS name resolution. These attacks are similar to neighbor discovery attacks and attacks against unauthenticated DHCP. We recommend using the Secure Neighbor Discovery (SEND) protocol as a security mechanism for neighbor discovery to allow all the neighbor discovery options including the recursive DNS server options to be automatically included in the signatures.

For more information about configuring the SEND protocol, see www.juniper.net/documentation/en_US/junos14.1/topics/topic-map/ipv6-secure-neighbor.html

RELATED DOCUMENTATION

Configuring a Recursive DNS Server Address for IPv6 Hosts

For IPv6 hosts, a maximum of three recursive DNS server addresses can be configured along with their respective lifetimes. The configured IPv6 host uses the specified recursive DNS server address for DNS resolution where the IPv6 host's address is autoconfigured through an IPv6 stateless address and where there is no DHCPv6 infrastructure available.

To configure a recursive DNS server address on IPv6 hosts, follow these steps:

1. Configure the recursive DNS server address for the IPv6 host.

```
[edit protocols router-advertisement]
user@host# set interfaces interface name dns-server-address address
```

   For example, to assign IPv6 address abcd:1::1 as the recursive dns server address to interface fe-1/0/1:

```
[edit protocols router-advertisement]
user@host# set interfaces fe-1/0/1 dns-server-address abcd:1::1
```

2. Configure the lifetime to specify the time in seconds for which the recursive DNS server address remains valid.

```
[edit protocols router-advertisement interfaces interface name dns-server-address address]
user@host# set lifetime seconds
```

For example, to specify a lifetime of 60 seconds for the recursive DNS server address:

```
[edit protocols router-advertisement interfaces interface name dns-server-address address]
user@host# set lifetime 60
```

The default value of the lifetime of the configured recursive DNS server address is 1800 seconds.

RELATED DOCUMENTATION

- *dns-server-address*
- *lifetime*
- Understanding Recursive DNS Servers for IPv6 | 16
- Example: Configuring Recursive DNS Server Addresses for IPv6 Hosts | 18

Example: Configuring Recursive DNS Server Addresses for IPv6 Hosts

IN THIS SECTION

This example shows how to configure the recursive DNS server address of an IPv6 host. The recursive DNS server address is included in the router advertisement that is sent to the neighboring devices.

Requirements

This example uses the following hardware and software components:

- Two MX Series routers with IPv6 enabled on the connected interfaces.
- Junos OS Release 14.1 or later running on all devices.

Overview

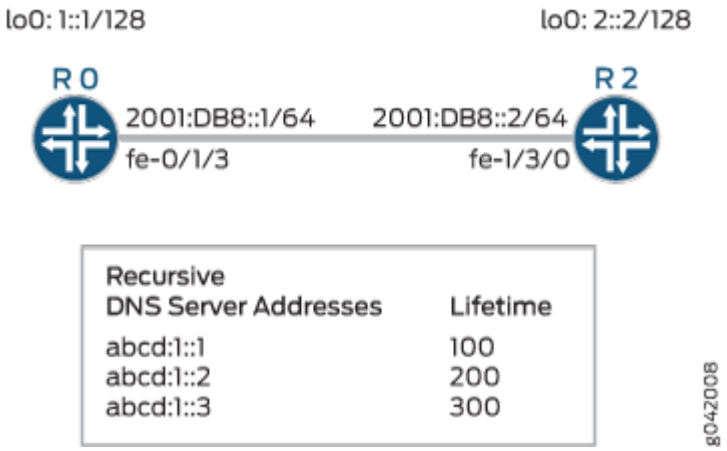
IN THIS SECTION

The example includes two routers that are directly connected. Configure IPv6 on the directly connected interfaces. Enable router advertisement on the interfaces and configure the recursive DNS server addresses and their lifetimes on the interfaces. This example verifies that the router advertisement sent to the neighboring device includes the configured recursive DNS server addresses.

Topology

Figure 2 on page 19 shows the sample topology.

Figure 2: Configuring Recursive DNS Server Address for IPv6 Hosts



Configuration

IN THIS SECTION

CLI Quick Configuration

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

Router R0

```
set interfaces fe-0/1/3 unit 0 family inet6 address 2001:DB8::1/64
set interfaces lo0 unit 0 family inet6 address 1::1/128
set protocols router-advertisement interface fe-0/1/3 max-advertisement-interval 4
set protocols router-advertisement interface fe-0/1/3 min-advertisement-interval 3
set protocols router-advertisement interface fe-0/1/3 dns-server-address abcd:1::1 lifetime 100
set protocols router-advertisement interface fe-0/1/3 dns-server-address abcd:1::2 lifetime 200
set protocols router-advertisement interface fe-0/1/3 dns-server-address abcd:1::3 lifetime 300
```

Router R1

```
set interfaces fe-1/3/0 unit 0 family inet6 address 2001:DB8::2/64
set interfaces lo0 unit 0 family inet6 address 1::2/128
set protocols router-advertisement interface fe-1/3/0 max-advertisement-interval 4
set protocols router-advertisement interface fe-1/3/0 min-advertisement-interval 3
set protocols router-advertisement interface fe-1/3/0 dns-server-address abcd:1::4 lifetime 100
```

Configuring the Recursive DNS Server Address

Step-by-Step Procedure

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

NOTE: Repeat this procedure for Router R1, modifying the appropriate interface names, addresses, and any other parameters for the router.

To configure the recursive DNS server address on Router R0:

1. Enable IPv6 on the physical interface.

```
[edit interfaces]
user@R0# set fe-0/1/3 unit 0 family inet6 address 2001:DB8::1/64
```

2. Configure the loopback address.

```
[edit interfaces]
user@R0# set lo0 unit 0 family inet6 address 1::1/128
```

3. Specify the time interval between router advertisements on the interface.

   The router sends advertisements to neighbors after the specified time interval. In this example, Router R0 sends router advertisements to Router R1 after a minimum interval of 3 seconds and a maximum interval of 4 seconds.

```
[edit protocols router-advertisement]
user@R0# set interface fe-0/1/3 max-advertisement-interval 4
user@R0# set interface fe-0/1/3 min-advertisement-interval 3
```

4. Configure the recursive DNS addresses and their lifetimes on the interface.

```
[edit protocols router-advertisement]
user@R0# set interface fe-0/1/3 dns-server-address abcd:1::1 lifetime 100
```

```
user@R0# set interface fe-0/1/3 dns-server-address abcd:1::2 lifetime 200
user@R0# set interface fe-0/1/3 dns-server-address abcd:1::3 lifetime 300
```

Results

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

```
user@R0# show interfaces
fe-0/1/3 {
    unit 0 {
        family inet6 {
            address 2001:DB8::1/64;
        }
    }
}
lo0 {
    unit 0 {
        family inet6 {
            address ::1/128;
        }
    }
}
user@R0# show protocols
router-advertisement {
    interface fe-0/1/3.0 {
        max-advertisement-interval 4;
        min-advertisement-interval 3;
        dns-server-address abcd:1::1 {
            lifetime 100;
        }
        dns-server-address abcd:1::2 {
            lifetime 200;
        }
        dns-server-address abcd:1::3 {
            lifetime 300;
        }
    }
}
```

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

```
user@R0# commit
```

Verification

IN THIS SECTION

Verifying That the Router Advertisement Includes the Recursive DNS Server Address

Purpose

Verify that the router advertisement on Router R1 includes the recursive DNS server address configured on Router R0.

Action

From operational mode on Router R1, enter the **show ipv6 router-advertisement** command.

```
user@R1> show ipv6 router-advertisement

Interface: fe-1/3/0.0
  Advertisements sent: 18, last sent 00:00:02 ago
  Solicits received: 0
  Advertisements received: 18
  Advertisement from fe80::214:f6ff:fe22:5422, heard 00:00:02 ago
    Managed: 0
    Other configuration: 0
    Reachable time: 0 ms
    Default lifetime: 12 sec
    Retransmit timer: 0 ms
    Current hop limit: 64
    RDNSS address: abcd:1::1
```

```
      Lifetime: 100 sec
    RDNSS address: abcd:1::2
      Lifetime: 200 sec
    RDNSS address: abcd:1::3
      Lifetime: 300 sec
```

Meaning

The recursive DNS server address and the configured lifetime are included in the router advertisements on Router R1.

RELATED DOCUMENTATION

Configuring a Recursive DNS Server Address for IPv6 Hosts | **17**

Understanding Recursive DNS Servers for IPv6 | **16**

dns-server-address

lifetime

4

CHAPTER

Monitoring ICMP Router Discovery

Traceroute for Inactive Interface

Traceroute is a tool for displaying the route taken by a packet from an IP network on their way to a given host. When a traceroute is performed the packets are always sent out of the interface that is the NH for the active route and there is no option to bypass it.

When a traceroute is performed, packets are sent out of active interface even if we specify an inactive interface. From Junos OS Release 17.4R1 onwards, you can configure traceroute to send out packets through an inactive next-hop by specifying the `traceroute` *`next-hop address`* to a destination through an inactive next hop.

RELATED DOCUMENTATION

traceroute

Example: Tracing Global Routing Protocol Operations

IN THIS SECTION

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

IN THIS SECTION

CLI Quick Configuration

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

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

Configuring Trace Operations

Step-by-Step Procedure

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

To configure the trace operations:

1. Configure trace operations.

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

```
user@host# set file files 10
user@host# set flag route
```

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

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

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

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

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

IN THIS SECTION

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
```

How to Use the Probe Command

SUMMARY

Learn how to configure and use the Probe command.

IN THIS SECTION

Benefits of the Probe command

The Probe command is used to display an interface's operational state, and to determine if IPv4, IPv6, or both address families are configured. Unlike a conventional ping, the probe command can obtain the operational state of an interface for which the probing node does not have a route or a shared address family. For example, An IPv4-only node can use the Probe command to determine the operational state of an IPv6 address on the probed node.

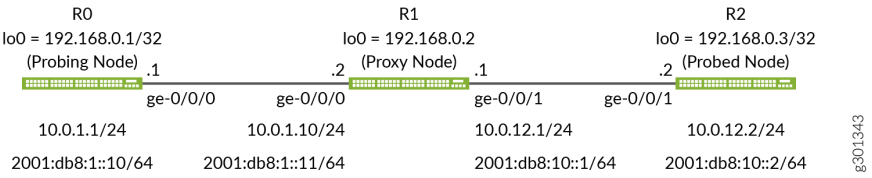
What is the Probe Command?

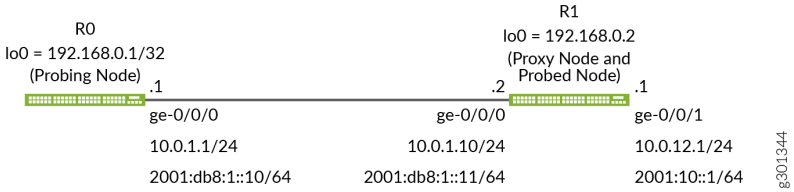
IN THIS SECTION

RFC 8335, *PROBE: A Utility for Probing Interfaces*, describes the Probe utility. Probe is a network diagnostic tool similar to Ping that can be used to query the status of a probed interface on a node. The Probe command require bidirectional connectivity between the probing interface and the proxy interface. The proxy interface can reside on the same node as the probed interface, or it can reside on a node to which the probed interface is directly connected. Probe uses ICMP Extended Echo/Reply messages for communication between the probing interface and the proxy interface. This utility helps in scenarios where bidirectional connectivity between the probing and probed interfaces is lacking. For instance, if the probed interface is an unnumbered interface, or if the probed interface is assigned a different address family. In both cases you can use the Probe command to confirm if the probed

interface is reachable from the proxy. The proxy interface learns details about the probed interface by inspecting its local ARP and NDP entries.

The first figure shows R0 as the probing node, R1 as the proxy node, and R2 as the probed node. In the second figure the R1 node functions in both the proxy and probed node roles:





Enabling the Probe command

To enable the `probe` command, configure the `extended-echo` configuration statement at the `[edit system]` hierarchy level on the proxy node.

```
[edit]
user@host# set system extended-echo
```

NOTE: You do not need to enable the `extended-echo` configuration statement on the probing or probed nodes.

Using the Probe command

You can probe using the remote address of the probed device when the proxy and probed nodes are not the same. That is, when the proxy and the probed nodes are two different nodes. The proxy interface learns the details of the probed interface from its local ARP or NDP entries associated with the probed node.

You can also probe using the following three options when the proxy and probed nodes are same:

- By using the IP address

- By using the interface name
- By using the interface index

NOTE: 1. You can probe for information with any one of these mandatory options. That is, `by-remote-address`, `by-address`, `by-name`, or `by-index`.

2. You can probe using the IPv4 or IPv6 address to query for the status of the IPv4 address. Likewise, you can probe using the IPv4 or IPv6 address to query for the status of the IPv6 address.

The following examples shows how to use the `probe` command.

Purpose

Use case 1: Probing when the proxy interface and the probed interface reside on two different nodes (proxy node (R1) and the probed node (R2)). Consider R0 as the probing node, R1 as the proxy node, and R2 as the probed node and change the details of the interfaces and IP addresses to match your network configuration. In this case, you can probe using `by-remote-address` option only.

Probe for the status of a remote IP address using a proxy interface.

Action

Example 1: From operational mode, probe for the status of the IP address 10.0.12.2 of the probed node (R2) using the `by-remote-address` option with the proxy IP address 10.0.1.10 of the proxy node (R1).

In this case, the proxy node provides the status of the probed interface based on the information associated with its local ARP and NDP entries. Use the `count` argument to control the number of probe request that are sent.

NOTE:

- In this example, the target address (10.0.12.2) is defined on the probed node. As a result, you can see a positive result, which confirms that the IP address is reachable on the probed node.
- When you probe using the `by-remote-address` option, you can only determine if the probed interface is reachable. That is, if the probed interface is in reachable state, then by default, the probe packet statistics such as active, IPv4, and IPv6 are set to zero (0).

user@R0>**probe 10.0.1.10 by-remote-address 10.0.12.2 count 1**

```
PROBE 10.0.1.10 (10.0.1.10):
32 bytes from 10.0.1.10: icmp_seq=0 ttl=255 code=0 state=2 active=0 IPv4=0 IPv6=0 time=6.048 ms

--- 10.0.1.10 probe statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.048/6.048/6.048/0.000 ms
```

Meaning

The proxy IP address (10.0.1.10) displays the status of the probed IP address (10.0.12.2). The output is verified with the following probe packet statistics:

- `code=0` indicates that there is no error.
- `state=2` indicates that the probed IP address 10.0.12.2 is reachable.
- `active=0` is set to zero and ignored upon receipt when you probe using the `by-remote-address` option.
- `IPv4=0` and `IPv6=0` by default when you probe using the `by-remote-address`.
- `time=`*`milliseconds`* `ms` indicates the time taken to receive the reply after the request is transmitted.
- `ttl` is the IPv4 time to live value, which default to the maximum value.

Example 2: From operational mode, probe for the status of the IP address 10.0.12.22 of the probed node (R2) using the `by-remote-address` option with the proxy IP address 10.0.1.10 of the proxy node (R1).

As in Example 1, the proxy node provides the status of the probed interface based on the information associated with its local ARP and NDP entries. Use the `count` argument to control the number of probe request that are sent.

NOTE:

- In this example the target address (10.0.12.22) is not defined on the probed node. As a result, you expect to see a negative result, which confirms the IP address is not active on the probed node.

- When you probe using the `by-remote-address` option, you can only determine if the probed interface is reachable. That is, if the probed interface is in reachable state, then by default, the probe packet statistics such as active, IPv4, and IPv6 are set to zero (0).

user@R0>**probe 10.0.1.10 by-remote-address 10.0.12.22 count 1**

```
PROBE 10.0.1.10 (10.0.1.10):
32 bytes from 10.0.1.10: icmp_seq=0 ttl=255 code=3 state=0 active=0 IPv4=0 IPv6=0 time=5.054 ms

--- 10.0.1.10 probe statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.054/5.054/5.054/0.000 ms
```

Meaning

The proxy IP address (10.0.1.10) displays the status of the probed IP address (10.0.12.22). The output is verified with the following probe packet statistics:

- `code=3` indicates that there is no such entry in the ARP table. This is expected in this example because the probed IP address is *not* defined on either the proxy or probed nodes.
- `state=0` is a required setting when the `code` bit is set to a non-zero state and the probed interface does not reside on the proxy node. Here, the probed IP address resides on the probed node, and the `code` bit is set to a `3`.
- `active=0` is set to zero and ignored upon receipt when you probe using the `by-remote-address` option.
- `IPv4=0` and `IPv6=0` by default when you probe using the `by-remote-address`.
- `time=`*`milliseconds`* `ms` indicates the time taken to receive the reply after the request is transmitted.
- `ttl` is the IPv4 time to live value, which default to the maximum value.

Purpose

Use Case 2: Probing when proxy node and the probed node are the same. Consider R0 as the probing node, R1 as the proxy and the probed node. Change the details of the interfaces and IP addresses to match your network configuration.

Query for the status of the probed IP address through the proxy IP address

Action

From operational mode, probe for the status of the probed IP address 10.0.12.1 using the `by-address` option with the proxy IP address 10.0.1.10 at the proxy node (R1). The `count` argument is used to set the number of probe requests to 1.

user@R0>**probe 10.0.1.10 by-address 10.0.12.1 count 1**

```
PROBE 10.0.1.10 (10.0.1.10):
32 bytes from 10.0.1.10: icmp_seq=0 ttl=255 code=0 state=0 active=1 IPv4=1 IPv6=1 time=6.056 ms
--- 10.0.1.10 probe statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.056/8.928/13.459/3.242 ms
```

Meaning

The proxy IP address, 10.0.1.10 displays the status of the probed IP address, 10.0.12.1. The output is verified with the following probe packet statistics:

- `code=0` indicates that the Probe request completed without error.
- `state=0` indicates that the probed interface is active and resides on the proxy node.
- `active=1` indicates that the probed interface is active.
- `IPv4=1` and `IPv6=1` indicates that both IPv4 and IPv6 addresses are configured and available on the probed node.
- `time=`*`milliseconds`* `ms` indicates the time taken to receive the reply after the request is transmitted.
- `ttl` is the IP time to live value, which is set to the maximum hop count..

Purpose

Use Case 3: Query for the status of the IPv4 or IPv6 address using the interface index of the probed interface. (Probing when the proxy node and the probed node are the same. Consider R0 as the probing node, R1 as the proxy and the probed node. Change the details of the interfaces and IP addresses to match your network configuration.)

Action

From operational mode, probe for the status of the IPv4 or IPv6 address using the `by-index` option to specify the interface index of the probed interface. The `count` 2 argument causes 2 probe request to be generated. In this example the R1 node functions as both the proxy and probed nodes.

user@R0>**probe 10.0.1.10 by-index 333 count 2**

```
PROBE 10.0.1.10 (10.0.1.10):
28 bytes from 10.0.1.10: icmp_seq=0 ttl=255 code=0 state=0 active=1 IPv4=1 IPv6=1 time=6.767 ms
28 bytes from 10.0.1.10: icmp_seq=1 ttl=255 code=0 state=0 active=1 IPv4=1 IPv6=1 time=3.796 ms

--- 10.0.1.10 probe statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.796/5.282/6.767/1.485 ms
```

Meaning

The proxy IP address, 10.0.1.10 displays the status of the probed interface using the Logical interface index-id, 333. The output is verified with the following probe packet statistics:

- `code=0` indicates that the request is completed without error.
- `state=0` indicates that the probed interface is active and resides on the proxy node.
- `active=1` indicates that the probed interface is active.
- `IPv4=1` and `IPv6=1` indicates that IPv4 and IPv6 addresses are configured on the probed node.
- `time=`*`milliseconds`* `ms` indicates the time taken to receive the reply after the request is transmitted.
- `ttl` is the IP time to live value..

Purpose

Use Case 4: Query the proxy interface for information about the probed interface that reside on the proxy node. (Probing when the proxy node and the probed node are the same. Consider R0 as the probing node, R1 as the proxy and the probed node. Change the details of the interfaces and IP addresses to match your network configuration.)

Action

From operational mode, probe for the status of the probed IP address using the `by-address` option while specifying the proxy node by its IPv6 address. The `count` argument is used to set the probe count to 1.

user@R0>**probe 2001:db8:1::11 by-address 10.0.12.1 count 1**

```
PROBE6(72=40+8+24 bytes) 2001:db8:1::10 --> 2001:db8:1::11
32 bytes from 2001:db8:1::11, icmp_seq=0 hlim=255 code=0 state=0 active=1 IPv4=1 IPv6=1
time=6.443 ms

--- 2001:db8:1::11 probe6 statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 6.443/6.443/6.443/0.000 ms
```

Meaning

The proxy IP address, 2001:db8:1::11 displays the status of the probed IP address, 10.0.12.1 that resides on the proxy device. The output is verified with the following probe packet statistics:

- `code=0` indicates that the probe request completes with out error.
- `state=0` indicates that the probed interface is active and resides on the proxy node.
- `active=1` indicates that the probed interface is active.
- `IPv4=1` and `IPv6=1` indicates that both IPv4 and IPv6 addresses are configured on the probed interface.
- `time=`*`milliseconds`* `ms` indicates the time taken to receive the reply after the request is transmitted.
- `hlim` is the IPv6 hop-limit which defaults to the maximum value.

Purpose

Use case 5: Query for the status of the probed interface using the interface name. (Probing when the proxy node and the probed node are the same. Consider R0 as the probing node, R1 as the proxy and the probed node. Change the details of the interfaces and IP addresses to match your network configuration.)

Action

From operational mode, probe for the status of the probed interface using the interface name ge-0/0/1.0 using the IPv6 address of the proxy node 2001:db8:1::11.

user@R0> **probe 2001:db8:1::11 by-name ge-0/0/1.0 count 1**

```
PROBE6(74=40+8+26 bytes) 2001:db8:1::10 --> 2001:db8:1::11
34 bytes from 2001:db8:1::11, icmp_seq=0 hlim=255 code=0 state=0 active=1 IPv4=1 IPv6=1
time=4.750 ms
--- 2001:db8:1::10 probe6 statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 0.231/0.231/0.231/0.000 ms
```

Meaning

The proxy node with IPv6 address, 2001:db8:1::11 returns the status of the specified interface ge-0/0/1.0. The output is verified with the following probe packet statistics:

- `code=0` indicates that the probe request completes without error.
- `state=0` indicates that the probed interface is active on the proxy node.
- `active=1` indicates that the probed interface is active.
- `IPv4=1` and `IPv6=1` indicates that the IPv4 and IPv6 addresses are available on the probed interface.
- `time=`*`milliseconds`* `ms` indicates the time taken to receive the reply after the request is transmitted.
- `hlim` is the maximum IPv6 hop-limit-value.

SEE ALSO

> *extended-echo*
>
> *probe (ICMP Router Discovery)*

5
CHAPTER

Troubleshooting Network Issues

Working with Problems on Your Network

IN THIS SECTION

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 1: Checklist for Working with Problems on Your Network

Tasks	Command or Action
Isolating a Broken Network Connection	
1. *Identifying the Symptoms of a Broken Network Connection*	**ping (*ip-address* | *hostname*) show route (*ip-address* | *hostname*) traceroute (*ip-address* | *hostname*)**
1. *Isolating the Causes of a Network Problem*	`show < configuration | interfaces | protocols | route >`
1. *Taking Appropriate Action for Resolving the Network Problem*	`[edit] delete routing options static route` *`destination-prefix`* **commit and-quit show route *destination-prefix***

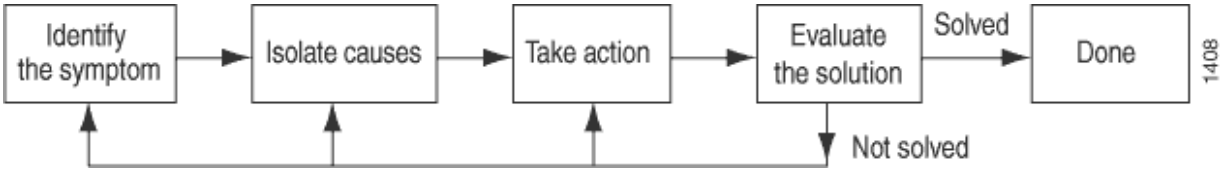
Table 1: Checklist for Working with Problems on Your Network *(Continued)*

Tasks	Command or Action
1. *Evaluating the Solution to Check Whether the Network Problem Is Resolved*	`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 3 on page 44, 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.

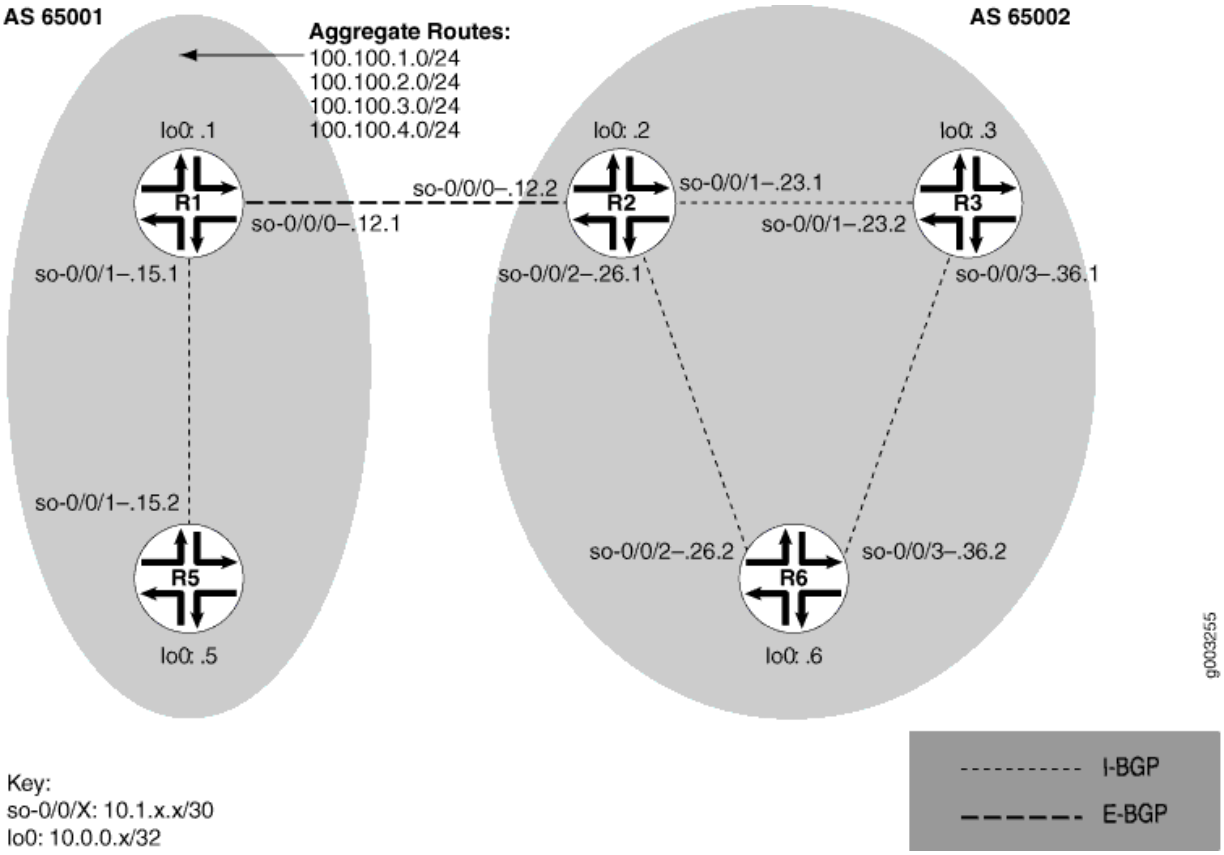
Figure 3: Process for Diagnosing Problems in Your Network



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 4 on page 45 shows the network topology used in this topic to illustrate the process of diagnosing problems in a network.

Figure 4: Network with a Problem



The network in Figure 4 on page 45 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*
- *Taking Appropriate Action for Resolving the Network Problem*
- *Taking Appropriate Action for Resolving the Network Problem*
- *Evaluating the Solution to Check Whether the Network Problem Is Resolved*

Identifying the Symptoms of a Broken Network Connection

IN THIS SECTION

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

IN THIS SECTION

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

IN THIS SECTION

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

IN THIS SECTION

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*, 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).

CHAPTER 6

Configuration Statements and Operational Commands

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Learn about the syntax and options that make up the statements and commands and understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements
- Operational Commands

7

CHAPTER

Knowledge Base