JUNIPER NETWORKS | Engineering Simplicity

Network Configuration Example

Configuring IP Monitoring on an SRX Series Device for the Branch

Published
2023-10-16

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

Network Configuration Example Configuring IP Monitoring on an SRX Series Device for the Branch

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

This document discusses the IP monitoring with route failover and IP monitoring with interface failover features. It also provides step-by-step examples of how to configure real-time performance monitoring, IP monitoring with route failover, IP monitoring with interface failover, IP monitoring with a DHCP backup interface, and IP monitoring within a virtual router.

This document focuses on Juniper Networks® SRX Series Services Gateways for the branch.

CHAPTER 1

Configuring IP Monitoring on an SRX Series Device for the Branch

About This Network Configuration Example

This document discusses the IP monitoring with route failover and IP monitoring with interface failover features. It also provides step-by-step examples of how to configure real-time performance monitoring, IP monitoring with route failover, IP monitoring with interface failover, IP monitoring with a DHCP backup interface, and IP monitoring within a virtual router.

This document focuses on Juniper Networks® SRX Series Services Gateways for the branch.

IP Monitoring Overview

IP monitoring is a technique that checks the reachability of an IP address or a set of IP addresses and takes an action when the IP address is not reachable. The action that IP monitoring takes can be one of the following:

- Add a new route that has a higher priority (lower preference) value than a route configured through the CLI.
- Enable a backup interface.
- Add a weight to a redundancy group.

After one of these actions is taken, reachability probes are executed. When the probes begin to succeed, the action taken by IP monitoring is reversed.

IP monitoring is supported on all branch SRX Series devices from the SRX100 to the SRX650 running the following software release:

- IP monitoring with route failover is supported in Juniper Networks Junos® operating system (Junos OS) Release 11.2 R2 and later.
- IP monitoring with interface failover is supported for high availability on Junos OS Release 11.4 R2 and later.

For information about IP monitoring for high availability, see: https://www.juniper.net/documentation/en_US/junos12.1x45/topics/example/chassis-cluster-redundancy-group-ip-address-monitoring-configuring-cli.html. IP monitoring for high availability is not discussed in this document.

IP monitoring is a very handy tool for automatic backup scenarios. It enhances backup WAN connectivity. It can be used:

- If you have an expensive or pay-per-bit backup link such as 3G/4G LTE and want to enable it only when reachability through the primary WAN link fails.
- If performance parameters including latency and jitter are affecting the primary WAN connection, and you want to automatically select a backup WAN connection to avoid possible performance degradation.
- If you have two routes to a destination but want to use a specific route only if the path to the destination through the other route is not available.

> **NOTE**: There are circumstances in which two routes are present, but the destination is not reachable if you pick a particular route.

The following limitations exist with IP monitoring:

- With IP monitoring with route failover, you do not have the ability to specify the preference value for a route.
- Also, you do not have the ability to stop the fail-back of a backup interface or an injected route back to the primary interface or route. If the reachability of the monitored IP address flaps, you might get into a scenario where the system keeps flapping between the primary and backup route or interface. Currently IP monitoring with route failover does not have the ability to stop this.
- The backup interface for IP monitoring with interface failover cannot be a secure tunnel interface. However, the IPsec VPN feature has a feature called *VPN monitoring* which accomplishes the same thing.

RELATED DOCUMENTATION

Configuring Real-Time Performance Monitoring Probes

For both IP monitoring with route failover and IP monitoring with interface failover, the probes that are used to test the target device are real-time performance monitoring (RPM) probes that not only test the reachability of the IP address but also perform service-level monitoring on parameters such as jitter and latency.

Real-time performance monitoring allows you to perform service-level monitoring. When RPM is configured on a device, the device calculates network performance based on packet response time, jitter, and packet loss. These values are gathered by HTTP GET requests, Internet Control Message Protocol (ICMP) requests, TCP requests, and UDP requests, depending on the configuration.

You can gather RPM statistics by sending out probes to a specified probe target, identified by an IP address or URL. When the target receives the probe, it generates responses, which are received by the SRX Series device. By analyzing the transit times to and from the remote server, the device can determine network performance.

SRX Series gateways send out the following probe types:

- HTTP GET request to a target URL
- HTTP GET request for metadata from a target URL
- ICMP echo request to a target IP address (the default)
- ICMP timestamp request to a target address
- UDP ping packets to a target device
- UDP timestamp requests to a target IP address
- TCP ping packets to a target device

NOTE: The default probe is an ICMP echo request unless otherwise configured.

Each probed target is monitored over the course of a test. A test represents a collection of probes, sent out at regular intervals as defined in the configuration. Statistics are then returned for each test. Because a test is a collection of probes that have been monitored over some amount of time, test statistics such as standard deviation and jitter can be calculated and included with the average probe statistics.

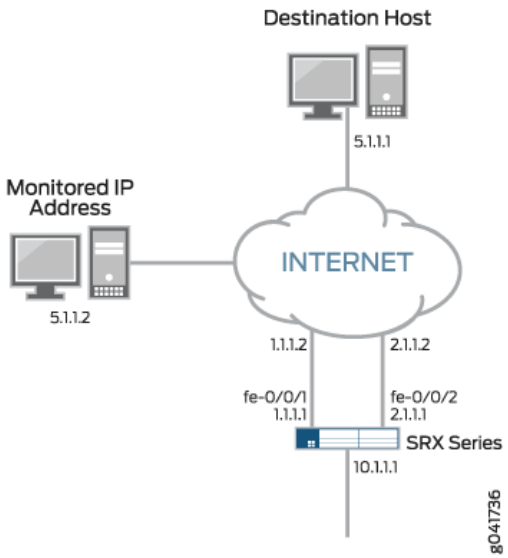
Within a test, RPM probes are sent at regular intervals and configured in seconds. When the total number of probes has been sent and the corresponding responses received, the test is complete. You can manually set the probe interval for each test to control how the RPM test is conducted.

After all probes for a particular test have been sent, the test begins again. The time between tests is the test interval. You can manually set the test interval to tune RPM performance.

To monitor multiple IP addresses, multiple tests can be defined for each probe, and the probe fails only if all tests fail. The system can perform a *logical AND* operation of all test results to determine the outcome of a probe.

Figure 1 on page 5 shows the topology used in the following configuration example.

Figure 1: Real-Time Performance Monitoring Topology



The following is an example configuration of an RPM probe that monitors three IP addresses:

```
set services rpm probe Probe-Payment-Server test paysvr target address 5.1.1.3
set services rpm probe Probe-Payment-Server test paysvr probe-count 5
set services rpm probe Probe-Payment-Server test paysvr probe-interval 5
set services rpm probe Probe-Payment-Server test paysvr test-interval 3
set services rpm probe Probe-Payment-Server test paysvr thresholds successive-loss 5

set services rpm probe Probe-Payment-Server test paysvr1 target address 5.1.1.2
set services rpm probe Probe-Payment-Server test paysvr1 probe-count 5
set services rpm probe Probe-Payment-Server test paysvr1 probe-interval 5
set services rpm probe Probe-Payment-Server test paysvr1 test-interval 3
```

```
set services rpm probe Probe-Payment-Server test paysvr1 thresholds successive-loss 5

set services rpm probe Probe-Payment-Server test paysvr2 target address 5.1.1.5
set services rpm probe Probe-Payment-Server test paysvr2 probe-count 5
set services rpm probe Probe-Payment-Server test paysvr2 probe-interval 5
set services rpm probe Probe-Payment-Server test paysvr2 test-interval 3
set services rpm probe Probe-Payment-Server test paysvr2 thresholds successive-loss 5
```

After the RPM probe configuration is committed, results of the probe can be displayed using the following command:

```
root# run show services rpm probe-results

   Owner: Probe-Payment-Server, Test: paysvr
   Target address: 5.1.1.3, Probe type: icmp-ping
   Destination interface name: fe-0/0/1.0
   Test size: 5 probes
   Probe results:
       Request timed out, Tue Sep 20 02:22:28 2011
   Results over current test:
       Probes sent: 1, Probes received: 0, Loss percentage: 100
   Results over last test:
       Probes sent: 5, Probes received: 0, Loss percentage: 100
   Results over all tests:
       Probes sent: 56, Probes received: 0, Loss percentage: 100

   Owner: Probe-Payment-Server, Test: paysvr1
   Target address: 5.1.1.2, Probe type: icmp-ping
   Destination interface name: fe-0/0/1.0
   Test size: 5 probes
   Probe results:
       Response received, Tue Sep 20 02:22:27 2011, No hardware timestamps
Rtt: 1742 usec
   Results over current test:
       Probes sent: 2, Probes received: 2, Loss percentage: 0
     Measurement: Round trip time
    Samples: 2, Minimum: 1582 usec, Maximum: 1742 usec, Average: 1662 usec,
    Peak to peak: 160 usec, Stddev: 80 usec, Sum: 3324 usec
Results over last test:
    Probes sent: 5, Probes received: 5, Loss percentage: 0
    Test completed on Tue Sep 20 02:22:19 2011
```

```
      Measurement: Round trip time
    Samples: 2, Minimum: 1582 usec, Maximum: 1742 usec, Average: 1662 usec,
    Peak to peak: 160 usec, Stddev: 80 usec, Sum: 3324 usec
Results over last test:
    Probes sent: 5, Probes received: 5, Loss percentage: 0
    Test completed on Tue Sep 20 02:22:19 2011
    Measurement: Round trip time
        Samples: 5, Minimum: 1454 usec, Maximum: 1701 usec, Average: 1587 usec,
        Peak to peak: 247 usec, Stddev: 92 usec, Sum: 7935 usec
Results over all tests:
    Probes sent: 67, Probes received: 67, Loss percentage: 0
    Measurement: Round trip time
        Samples: 67, Minimum: 1427 usec, Maximum: 712721 usec,
        Average: 13074 usec, Peak to peak: 711294 usec, Stddev: 86142 usec,
        Sum: 875977 usec

   Owner: Probe-Payment-Server, Test: paysvr2
   Target address: 5.1.1.5, Probe type: icmp-ping
   Destination interface name: fe-0/0/1.0
   Test size: 5 probes
   Probe results:
       Request timed out, Tue Sep 20 02:22:28 2011
   Results over current test:
       Probes sent: 1, Probes received: 0, Loss percentage: 100
   Results over last test:
       Probes sent: 5, Probes received: 0, Loss percentage: 100
   Results over all tests:
       Probes sent: 56, Probes received: 0, Loss percentage: 100
```

With the RPM probes, you can detect the reachability of the monitored IP address, and you can also measure network parameters and take an action if round-trip time (RTT) or jitter is greater than a configured value. The following command displays the parameters that can be measured:

```
root# set services rpm probe probetoremote test paysrvr thresholds ?

Possible completions:
  <[Enter]>            Execute this command
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  egress-time          Maximum source to destination time per probe
  ingress-time         Maximum destination to source time per probe
  jitter-egress        Maximum source to destination jitter per test
```

```
jitter-ingress      Maximum destination to source jitter per test
jitter-rtt          Maximum jitter per test (0..60000000 microseconds)
rtt                 Maximum round trip time per probe (microseconds)
std-dev-egress      Maximum source to destination standard deviation per test
std-dev-ingress     Maximum destination to source standard deviation per test
std-dev-rtt         Maximum standard deviation per test (microseconds)
successive-loss     Successive probe loss count indicating probe failure
total-loss          Total probe loss count indicating test failure (0..15)
|                   Pipe through a command
```

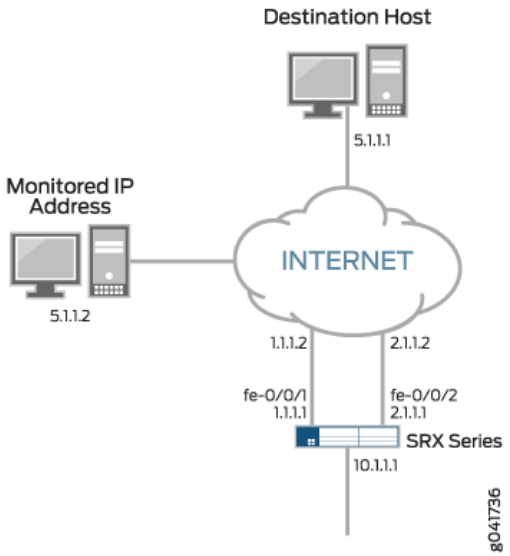
RELATED DOCUMENTATION

Configuring IP Monitoring with Route Failover

Using IP monitoring with route failover, you can track an IP address or a set of IP addresses using a real-time performance monitoring (RPM) probe. If the RPM probe fails, you can inject a route into the routing table. After the RPM probe successfully reaches its target, the route is withdrawn from both the routing and forwarding tables.

Figure 2 on page 9 shows the topology used in the configuration example and how IP monitoring works.

Figure 2: Real-Time Performance Monitoring Topology



In the normal state of operation, the next-hop router to reach IP address 5.1.1.1 on the SRX Series gateway is 1.1.1.2. However, when the RPM probe to IP address 5.1.1.2 fails, you should use IP address 2.1.1.2 as the next hop.

To achieve this result, define an RPM probe to monitor IP address 5.1.1.2. Enter the following configuration:

```
set services rpm probe Probe-Payment-Server test paysvr target address 5.1.1.2
set services rpm probe Probe-Payment-Server test paysvr probe-count 5
set services rpm probe Probe-Payment-Server test paysvr probe-interval 5
set services rpm probe Probe-Payment-Server test paysvr test-interval 3
set services rpm probe Probe-Payment-Server test paysvr thresholds successive-loss 5
set services rpm probe Probe-Payment-Server test paysvr destination-interface fe-0/0/1.0
set services rpm probe Probe-Payment-Server test paysvr hardware-timestamp
set services rpm probe Probe-Payment-Server test paysvr next-hop 1.1.1.2
```

Also configure the IP monitoring policy to add a preferred route when the RPM probe fails. Enter the following configuration:

```
set services ip-monitoring policy payment match rpm-probe Probe-Payment-Server
set services ip-monitoring policy payment then preferred-route route 5.1.1.0/24 next-hop 2.1.1.2
```

In the steady state, you can reach IP address 5.1.1.1 through the device with the IP address 1.1.1.2, and the RPM probes are successful. To verify the operation of the steady state, use the following commands:

```
root# run traceroute 5.1.1.1 source 10.1.1.1

traceroute to 5.1.1.1 (5.1.1.1) from 10.1.1.1, 30 hops max, 40 byte packets
  1 1.1.1.2 (1.1.1.2) 14.697 ms 2.953 ms 9.043 ms
  2 5.1.1.1 (5.1.1.1) 9.916 ms 3.612 ms 4.085 ms
```

In the following `show` command output, the `PASS` results in the `Status` field indicates that the probe is successful:

```
root# run show services ip-monitoring status

Policy - payment
RPM Probes:
Probe name                 Address      Status
---------------------- ---------------- ---------
Probe-Payment-Server       5.1.1.2      PASS
Route-Action:
route-instance         route            next-hop       State
----------------- ----------------- ---------------- -------------
inet.0                5.1.1.0          2.1.1.2      NOT-APPLIED
```

In the following `show` command output, the `Probes sent` count and `Probes received` count are equal and the `Loss percentage` is `0`. This indicates that the probe is successful.

```
root# run show services rpm probe-results

    Owner: Probe-Payment-Server, Test: paysvr
    Target address: 5.1.1.2, Probe type: icmp-ping
    Destination interface name: fe-0/0/1.0
    Test size: 5 probes
    Probe results:
        Response received, Tue Sep 20 06:18:00 2011, No hardware timestamps
        Rtt: 1776 usec
    Results over current test:
        Probes sent: 5, Probes received: 5, Loss percentage: 0
        Measurement: Round trip time
            Samples: 5, Minimum: 1490 usec, Maximum: 7399 usec, Average: 2952 usec,
            Peak to peak: 5909 usec, Stddev: 2235 usec, Sum: 14758 usec
```

```
    Results over last test:
        Probes sent: 5, Probes received: 5, Loss percentage: 0
        Test completed on Tue Sep 20 06:18:00 2011
        Measurement: Round trip time
            Samples: 5, Minimum: 1490 usec, Maximum: 7399 usec, Average: 2952 usec,
            Peak to peak: 5909 usec, Stddev: 2235 usec, Sum: 14758 usec
    Results over all tests:
        Probes sent: 45, Probes received: 45, Loss percentage: 0
        Measurement: Round trip time
            Samples: 45, Minimum: 1490 usec, Maximum: 93350 usec,
            Average: 4766 usec, Peak to peak: 91860 usec, Stddev: 13517 usec,
            Sum: 214456 usec
```

When IP address 5.1.1.2 is unreachable, the RPM probes fail, and the route specified in the IP monitoring configuration is pushed to the routing table. The route pushed has a preference of one (1), which has a higher preference than any static route or route learned through a routing protocol. The server with the IP address of 5.1.1.1 is now reachable through the device with the IP address of `2.1.1.2`. To verify the operation of the fail state, use the following commands:

```
root# run show services ip-monitoring status

Policy - test-remote-server
    RPM Probes:
        Probe name                Address       Status
        --------------------- ---------------- ---------
        Probe-Payment-Server      5.1.1.2        FAIL
    Route-Action:
        route-instance         route            next-hop        State
        ---------------- ----------------- ---------------- -------------
        inet.0               5.1.1.0          2.1.1.2         APPLIED
```

In the following `show` command output, `to 2.1.1.2 via fe-0/0/2.0` indicates that the route has changed:

```
root# run show route 5.1.1.1

inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
5.1.1.0/24        *[Static/1] 00:00:18, metric2 0
                    > to 2.1.1.2 via fe-0/0/2.0
                    [Static/5] 00:01:38
                    > to 1.1.1.2 via fe-0/0/1.0
```

In the following `show` command output, `(2.1.1.2)` indicates that the route has changed from `(1.1.1.2)` shown in the steady state traceroute:

```
root# run traceroute 5.1.1.1 source 10.1.1.1
traceroute to 5.1.1.1 (5.1.1.1) from 10.1.1.1, 30 hops max, 40 byte packets
  1 2.1.1.2 (2.1.1.2) 9.436 ms 9.457 ms 9.011 ms
  2 5.1.1.1 (5.1.1.1) 3.671 ms 3.553 ms 4.036 ms
```

When IP address 5.1.1.2 is again reachable, the RPM probe successfully reaches its target, and the route that was added in the routing table is withdrawn.

To verify the operation of the restored steady state, use the following commands and verify that the results are similar to the steady-state results previously described:

```
root# run show services rpm probe-results

    Owner: Probe-Payment-Server, Test: paysvr
    Target address: 5.1.1.2, Probe type: icmp-ping
    Destination interface name: fe-0/0/1.0
    Test size: 5 probes
    Probe results:
        Response received, Tue Sep 20 08:00:02 2011, No hardware timestamps
        Rtt: 1410 usec
    Results over current test:
        Probes sent: 3, Probes received: 3, Loss percentage: 0
        Measurement: Round trip time
            Samples: 3, Minimum: 1410 usec, Maximum: 1769 usec, Average: 1596 usec,
            Peak to peak: 359 usec, Stddev: 147 usec, Sum: 4788 usec
    Results over last test:
        Probes sent: 5, Probes received: 5, Loss percentage: 0
        Test completed on Tue Sep 20 07:59:49 2011
        Measurement: Round trip time
            Samples: 5, Minimum: 1509 usec, Maximum: 3057 usec, Average: 1922 usec,
            Peak to peak: 1548 usec, Stddev: 579 usec, Sum: 9612 usec
    Results over all tests:
        Probes sent: 143, Probes received: 25, Loss percentage: 82
        Measurement: Round trip time
```

```
            Samples: 25, Minimum: 1410 usec, Maximum: 8086 usec, Average: 2973 usec,
            Peak to peak: 6676 usec, Stddev: 2337 usec, Sum: 74333 usec
```

```
root# run show route 5.1.1.1

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

5.1.1.0/24         *[Static/5] 00:13:18
                    > to 1.1.1.2 via fe-0/0/1.0
```

```
root# run show services ip-monitoring status

Policy - test-remote-server
    RPM Probes:
      Probe name           Address       Status
---------------------- ---------------- ---------
Probe-Payment-Server        5.1.1.2       PASS
Route-Action:
  route-instance         route           next-hop        State
----------------- ----------------- ---------------- -------------
    inet.0              5.1.1.0          2.1.1.2       NOT-APPLIED
```

```
root# run traceroute 5.1.1.1 source 10.1.1.1

traceroute to 5.1.1.1 (5.1.1.1) from 10.1.1.1, 30 hops max, 40 byte packets
1 1.1.1.2 (1.1.1.2) 9.590 ms 9.968 ms 15.589 ms
2 5.1.1.1 (5.1.1.1) 9.175 ms 3.914 ms 3.750 ms
```

It is important to note that in the RPM configuration, you specify the next-hop value. This guarantees that all of the probes (even after the failover) take the same route to reach the tracked IP address.

Without the next-hop value, it is possible that after the new route is injected (when the RPM probe fails), there might be a new route to reach the tracked IP address. It is also possible that if the system chooses this new route, an upstream router might not have a route to the tracked IP address, the probes might always fail, and the system might never fail back. Hence, it is always a best practice to include the `next-hop` statement in the configuration.

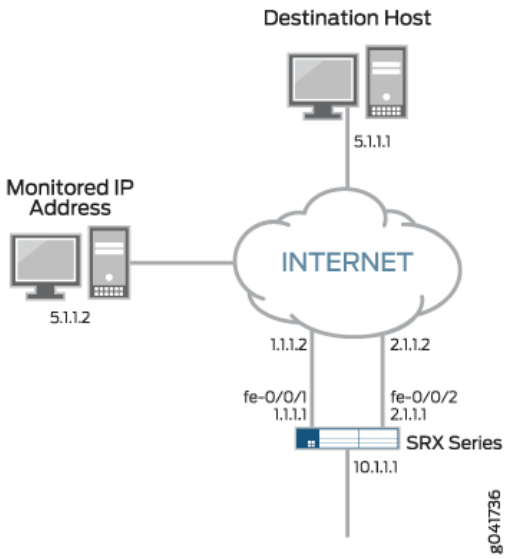
RELATED DOCUMENTATION

Configuring IP Monitoring with Interface Failover

Using IP monitoring with interface failover, you can track an IP address or a set of IP addresses using a real-time performance monitoring (RPM) probe. If the RPM probe fails, you can enable a backup interface that is normally down in the steady state. After the RPM probe successfully reaches its target, the backup interface is again disabled.

Figure 3 on page 14 shows the topology used in the configuration example and how IP monitoring works.

Figure 3: Real-Time Performance Monitoring Topology



In the steady state, the interface fe-0/0/2 is in the link down state. However, when the RPM probes fail, the system enables the interface, and traffic flows through interface fe-0/0/2. When the RPM probes

successfully reach their target, the system brings down interface fe-0/0/2 and traffic passes through fe-0/0/1.

To achieve this result, define an RPM probe to monitor IP address 5.1.1.2. Enter the following configuration:

```
set services rpm probe Probe-Payment-Server test paysvr target address 5.1.1.2
set services rpm probe Probe-Payment-Server test paysvr probe-count 5
set services rpm probe Probe-Payment-Server test paysvr probe-interval 5
set services rpm probe Probe-Payment-Server test paysvr test-interval 3
set services rpm probe Probe-Payment-Server test paysvr thresholds successive-loss 5
set services rpm probe Probe-Payment-Server test paysvr destination-interface fe- 0/0/1.0
set services rpm probe Probe-Payment-Server test paysvr hardware-timestamp
set services rpm probe Probe-Payment-Server test paysvr next-hop 1.1.1.2
```

Also configure the IP monitoring policy to enable the backup interface fe-0/0/2 when the RPM probe fails. Enter the following configuration:

```
set services ip-monitoring policy test-remote-server match rpm-probe Probe- Payment-Server
set services ip-monitoring policy test-remote-server then interface fe-0/0/2 enable
```

In this example, the interface fe-0/0/2 has a static IP address. Hence, you want to specify static routes to all destinations such that fe-0/0/2 is always the preferred route (lower preference value). You also need to specify routes to all destinations such that the next-hop router points to the next hop of fe-0/0/1 (higher preference value). With this approach, when interface fe-0/0/2 is disabled in the steady state, all traffic flows through fe-0/0/1. When interface fe-0/0/2 is up, all traffic flows through fe-0/0/2. Enter the following configuration:

```
set routing-options static route 5.1.1.0/24 qualified-next-hop 2.1.1.2 metric 1
set routing-options static route 5.1.1.0/24 qualified-next-hop 1.1.1.2 metric 10
```

If the backup interface has an IP address that is assigned using the Dynamic Host Configuration Protocol (DHCP), you do not know what the next hop through the backup interface is, and you are not able to add static routes similar to what is described previously. For more information about this scenario, see "Configuring IP Monitoring with a DHCP Backup Interface" on page 20.

In the steady state, you can reach IP address 5.1.1.1 through the link with the IP address of 1.1.1.2, and the RPM probes are successful. The backup interface fe-0/0/2 is down. To verify the steady state, enter the following command:

```
root# run traceroute 5.1.1.1 source 10.1.1.1

traceroute to 5.1.1.1 (5.1.1.1) from 10.1.1.1, 30 hops max, 40 byte packets
1 1.1.1.2 (1.1.1.2) 8.807 ms 14.808 ms 9.279 ms
2 5.1.1.1 (5.1.1.1) 3.517 ms 9.609 ms 3.804 ms
```

In the following `show` command output, the `PASS` results in the `Status` field indicate that the probe is successful:

```
root# run show services ip-monitoring status

Policy - test-remote-server
    RPM Probes:
      Probe name            Address      Status
---------------------- ---------------- ---------
Probe-Payment-Server        5.1.1.2        PASS

root# run show interfaces fe-0/0/2 terse

Interface      Admin Link Proto        Local         Remote
fe-0/0/2            down down
fe-0/0/2.0          up  down inet    2.1.1.1/24
```

In the following `show` command output, the `Probes sent` count and `Probes received` count are equal, and the `Loss percentage` is `0`. This indicates that the probe is successful.

```
root# run show services rpm probe-results

Owner: Probe-Payment-Server, Test: paysvr
Target address: 5.1.1.2, Probe type: icmp-ping
Destination interface name: fe-0/0/1.0
Test size: 5 probes
Probe results:
  Response received, Wed Sep 21 06:24:05 2011, No hardware timestamps
  Rtt: 1838 usec
Results over current test:
  Probes sent: 4, Probes received: 4, Loss percentage: 0
```

```
  Measurement: Round trip time
    Samples: 4, Minimum: 1674 usec, Maximum: 2006 usec, Average: 1805 usec,
    Peak to peak: 332 usec, Stddev: 132 usec, Sum: 7220 usec
Results over last test:
  Probes sent: 5, Probes received: 5, Loss percentage: 0
  Test completed on Wed Sep 21 06:23:47 2011
  Measurement: Round trip time
    Samples: 5, Minimum: 1632 usec, Maximum: 7599 usec, Average: 4226 usec,
    Peak to peak: 5967 usec, Stddev: 2719 usec, Sum: 21128 usec
Results over all tests:
  Probes sent: 54, Probes received: 54, Loss percentage: 0
  Measurement: Round trip time
    Samples: 54, Minimum: 1524 usec, Maximum: 97845 usec,
    Average: 5422 usec, Peak to peak: 96321 usec, Stddev: 13438 usec,
    Sum: 292762 usec
```

When IP address 5.1.1.2 is unreachable, the RPM probes fail and the interface fe-0/0/2 is enabled. All traffic is now routed through interface fe-0/0/2. The probes are still sent out of interface fe-0/0/1.

In the following `show` command output under the `Results over current test:` section, it shows the `Probes sent` count is `2` and the `Probes received` count is `0`. It also shows that the `Loss percentage` is `100`. This indicates that the probe has failed.

```
root# run show services rpm probe-results

Owner: Probe-Payment-Server, Test: paysvr
Target address: 5.1.1.2, Probe type: icmp-ping
Destination interface name: fe-0/0/1.0
Test size: 5 probes
Probe results:
  Request timed out, Thu Sep 22 01:18:25 2011
Results over current test:
  Probes sent: 2, Probes received: 0, Loss percentage: 100
Results over last test:
  Probes sent: 5, Probes received: 5, Loss percentage: 0
  Test completed on Thu Sep 22 01:18:17 2011
  Measurement: Round trip time
    Samples: 5, Minimum: 1635 usec, Maximum: 7528 usec, Average: 4055 usec,
    Peak to peak: 5893 usec, Stddev: 2819 usec, Sum: 20273 usec
Results over all tests:
  Probes sent: 22, Probes received: 20, Loss percentage: 9
  Measurement: Round trip time
```

```
    Samples: 20, Minimum: 1439 usec, Maximum: 9427 usec, Average: 3355 usec,
    Peak to peak: 7988 usec, Stddev: 2650 usec, Sum: 67099 usec
```

To further verify the fail state, use the following command:

```
root# run show services ip-monitoring status

Policy - test-remote-server
  RPM Probes:
    Probe name            Address        Status
---------------------- ---------------- ---------
Probe-Payment-Server      5.1.1.2         FAIL
```

To verify that interface fe-0/0/2 is enabled, use the following command:

```
root# run show interfaces fe-0/0/2 terse

Interface           Admin Link Proto          Local          Remote
fe-0/0/2               up up
fe-0/0/2.0             up up inet            2.1.1.1/24
```

To verify that IP address 5.1.1.1 is now reachable through the device with the IP address of 2.1.1.2, use the following command:

```
root# run traceroute 5.1.1.1 source 10.1.1.1

traceroute to 5.1.1.1 (5.1.1.1) from 10.1.1.1, 30 hops max, 40 byte packets
 1 2.1.1.2 (2.1.1.2) 9.031 ms 8.575 ms 15.450 ms
 2 5.1.1.1 (5.1.1.1) 10.120 ms 10.581 ms 3.553 ms
```

When IP address 5.1.1.2 is again reachable, the RPM probes successfully reach the target, and interface fe-0/0/2 is disabled. Now all traffic flows through interface fe-0/0/1.

To verify the operation of the restored steady state, use the following commands and verify that the results are similar to the steady-state results previously described:

```
root# run show services rpm probe-results

Owner: Probe-Payment-Server, Test: paysvr
Target address: 5.1.1.2, Probe type: icmp-ping
```

```
Destination interface name: fe-0/0/1.0
Test size: 5 probes
Probe results:
   Response received, Thu Sep 22 01:22:01 2011, No hardware timestamps
   Rtt: 2258 usec
Results over current test:
   Probes sent: 2, Probes received: 2, Loss percentage: 0
   Measurement: Round trip time
     Samples: 2, Minimum: 1847 usec, Maximum: 2258 usec, Average: 2053 usec,
     Peak to peak: 411 usec, Stddev: 206 usec, Sum: 4105 usec
Results over last test:
   Probes sent: 5, Probes received: 5, Loss percentage: 0
   Test completed on Thu Sep 22 01:21:53 2011
   Measurement: Round trip time
     Samples: 5, Minimum: 1614 usec, Maximum: 3752 usec, Average: 2213 usec,
     Peak to peak: 2138 usec, Stddev: 782 usec, Sum: 11064 usec
Results over all tests:
  Probes sent: 62, Probes received: 31, Loss percentage: 50
  Measurement: Round trip time
     Samples: 31, Minimum: 1439 usec, Maximum: 9427 usec, Average: 3076 usec,
     Peak to peak: 7988 usec, Stddev: 2426 usec, Sum: 95345 usec

root# run show services ip-monitoring status

Policy - test-remote-server
  RPM Probes:
    Probe name              Address      Status
---------------------- ---------------- ---------
Probe-Payment-Server        5.1.1.2       PASS

root# run show interfaces fe-0/0/2 terse

Interface        Admin Link Proto        Local           Remote
fe-0/0/2           down down
fe-0/0/2.0         up down inet        2.1.1.1/24
root# run traceroute 5.1.1.1 source 10.1.1.1

traceroute to 5.1.1.1 (5.1.1.1) from 10.1.1.1, 30 hops max, 40 byte packets
 1 1.1.1.2 (1.1.1.2) 8.818 ms 8.573 ms 9.847 ms
 2 5.1.1.1 (5.1.1.1) 3.384 ms 15.888 ms 3.640 ms
```

RELATED DOCUMENTATION

Configuring IP Monitoring with a DHCP Backup Interface

Many backup links like Ethernet and 3G/4G LTE links receive their IP address dynamically. This means that you might not know their IP address beforehand. Hence, it is hard to configure static routes such that the preferred route is through the backup interface (in the case of a real-time performance monitoring (RPM) probe failure). Also, most dynamic interfaces receive a 0.0.0.0/0 route from the DHCP server.

When a 0.0.0.0/0 route is installed in the routing table, it has a preference of 11. If the system has any static routes that are configured to use the primary WAN link, these routes have a preference of 5. Thus, if an RPM probe fails while using IP monitoring with interface failover and the backup interface is enabled, traffic might still be routed out of the primary interface through the static route (unless the interface is physically down). This is because the route has a preference of 5 which takes precedence over the default route the system obtained through DHCP (a preference of 11). If the system received its routes through a dynamic routing protocol, there is no problem as most routes have a preference value greater than 100. Hence, if the system has any static routes, we recommend adding a preference value greater than 12 to these static routes. In the "Configuring IP Monitoring with Interface Failover" on page 14 example, if fe-0/0/2 is a dynamic interface, you need to modify the static route configuration to reach IP subnet 5.1.1.0/24. To get this result, enter the following configuration:

```
set routing-options static route 5.1.1.0/24 next-hop 1.1.1.2
set routing-options static route 5.1.1.0/24 metric 1
set routing-options static route 5.1.1.0/24 preference 13
```

RELATED DOCUMENTATION

Configuring IP Monitoring with Interface Failover Using Boolean Conditions

Using IP monitoring with interface failover, it is possible to incorporate AND and OR conditions for monitored IP addresses. For the probe to fail, each test that makes up a given probe must fail (AND logic). If any of the probes specified for a given IP-monitoring policy fail then the IP-monitoring action is triggered (OR logic). The following example shows how to use rpm probes and IP-monitoring to implement AND and OR combinations. Suppose you want to bring up a backup interface when both IP addresses 11.1.1.1 and 11.1.1.2 are unreachable and also bring up the backup interface when IP address 12.1.1.1 is unreachable, then this scenario can be expressed as the following Boolean statement– “If (11.1.1.1 is unreachable AND 11.1.1.2 is unreachable) OR (12.1.1.1 is unreachable) then enable the backup interface”.

The configuration below implements the logic required for the scenario. The following commands show how the Junos features are substituted into in the above statement and map to the configuration. Probe1 and probe2 are defined in the example configuration below. Probe1 consists of two tests which are connected together with the boolean AND condition. If (probe1 fails OR probe2 fails) then enable the backup interface.

```
set services rpm probe probe1 test paysvr1 target address 11.1.1.1
set services rpm probe probe1 test paysvr1 probe-count 5
set services rpm probe probe1 test paysvr1 probe-interval 5
set services rpm probe probe1 test paysvr1 test-interval 3
set services rpm probe probe1 test paysvr1 thresholds successive-loss 5
set services rpm probe probe1 test paysvr1 destination-interface fe-0/0/1.0
set services rpm probe probe1 test paysvr1 hardware-timestamp
set services rpm probe probe1 test paysvr1 next-hop 1.1.1.2
set services rpm probe probe1 test paysvr2 target address 11.1.1.2
set services rpm probe probe1 test paysvr2 probe-count 5
```

```
set services rpm probe probe1 test paysvr2 probe-interval 5
set services rpm probe probe1 test paysvr2 test-interval 3
set services rpm probe probe1 test paysvr2 thresholds successive-loss 5
set services rpm probe probe1 test paysvr2 destination-interface fe-0/0/1.0
set services rpm probe probe1 test paysvr2 hardware-timestamp
set services rpm probe probe1 test paysvr2 next-hop 1.1.1.2
set services rpm probe probe2 test paysvr1 target address 12.1.1.1
set services rpm probe probe2 test paysvr1 probe-count 5
set services rpm probe probe2 test paysvr1 probe-interval 5
set services rpm probe probe2 test paysvr1 test-interval 3
set services rpm probe probe2 test paysvr1 thresholds successive-loss 5
set services rpm probe probe2 test paysvr1 destination-interface fe-0/0/1.0
set services rpm probe probe2 test paysvr1 hardware-timestamp
set services rpm probe probe2 test paysvr1 next-hop 1.1.1.2
set services ip-monitoring policy enable_backup match rpm-probe probe1
set services ip-monitoring policy enable_backup match rpm-probe probe2
set services ip-monitoring policy enable_backup then interface fe-0/0/2 enable
```

RELATED DOCUMENTATION

Configuring IP Monitoring in a Virtual Router

Real-time performance monitoring (RPM) probes can be sent from a virtual router. You can specify a route to be added in a virtual router from the IP monitoring configuration. You can also specify a backup interface to be enabled in a virtual router.

The following is a sample configuration for IP monitoring with route failover in a virtual router:

```
set services rpm probe Probe-Payment-Server test paysvr target address 5.1.1.2
set services rpm probe Probe-Payment-Server test paysvr probe-count 5
set services rpm probe Probe-Payment-Server test paysvr probe-interval 5
set services rpm probe Probe-Payment-Server test paysvr test-interval 3
set services rpm probe Probe-Payment-Server test paysvr routing-instance one
set services rpm probe Probe-Payment-Server test paysvr thresholds successive-loss 5
set services rpm probe Probe-Payment-Server test paysvr hardware-timestamp
set services ip-monitoring policy test-remote-server match rpm-probe Probe- Payment-Server
set services ip-monitoring policy test-remote-server then preferred-route routing-instances one
route 5.1.1.0/24 next-hop 2.1.1.2
set routing-instances one instance-type virtual-router
set routing-instances one interface fe-0/0/1.0
set routing-instances one interface fe-0/0/2.0
set routing-instances one interface fe-0/0/7.0
set routing-instances one routing-options static route 5.1.1.0/24 next-hop 1.1.1.2
```

The following is a sample configuration for IP monitoring with interface failover in a virtual router:

```
set services rpm probe Probe-Payment-Server test paysvr target address 5.1.1.2
set services rpm probe Probe-Payment-Server test paysvr probe-count 5
set services rpm probe Probe-Payment-Server test paysvr probe-interval 5
set services rpm probe Probe-Payment-Server test paysvr test-interval 3
set services rpm probe Probe-Payment-Server test paysvr hardware-timestamp
set services rpm probe Probe-Payment-Server test paysvr routing-instance one
set services rpm probe Probe-Payment-Server test paysvr thresholds successive-loss 5
set services ip-monitoring policy test-remote-server match rpm-probe Probe- Payment-Server
set services ip-monitoring policy test-remote-server then interface fe-0/0/2 enable
set routing-instances one instance-type virtual-router
set routing-instances one interface fe-0/0/1.0
set routing-instances one interface fe-0/0/2.0
set routing-instances one interface fe-0/0/7.0
set routing-instances one routing-options static route 5.1.1.0/24 qualified-nexthop 2.1.1.2
metric 1
set routing-instances one routing-options static route 5.1.1.0/24 qualified-nexthop 1.1.1.2
metric 10
set routing-instances one routing-options static route 5.1.1.2/32 next-hop 1.1.1.2
set routing-instances one routing-options static route 5.1.1.2/32 metric 1
set routing-instances one routing-options static route 5.1.1.2/32 preference 1
```

NOTE: When you define the RPM probe in a virtual router, you should not enter a value for the next-hop router at the `[edit set services rpm probe probe-owner test test-name]` hierarchy level. This is because the RPM daemon always looks in the `inet.0` route table for the next-hop address, even if you have specified a virtual router in the routing instance configuration. Because you cannot specify the next-hop router in the RPM configuration stanza, you need to add a /32 route with a preference of 1 to the tracked IP address so that the system always uses the route through the primary link for the RPM probe.

RELATED DOCUMENTATION

- IP Monitoring Overview | 2
- Configuring Real-Time Performance Monitoring Probes | 4
- Configuring IP Monitoring with Route Failover | 8
- Configuring IP Monitoring with Interface Failover | 14
- Configuring IP Monitoring with a DHCP Backup Interface | 20
- Configuring IP Monitoring with Interface Failover Using Boolean Conditions | 21