REAL-TIME PERFORMANCE MONITORING ON JUNIPER NETWORKS DEVICES

Tips and Tools for Assessing and Analyzing Network Efficiency
Table of Contents

Introduction: Understanding Real-Time Performance Monitoring ................................................. 1
Design Considerations .................................................................................................................. 1
Timestamps .................................................................................................................................... 2
Understanding Probes and Tests ................................................................................................. 4
Configuration and Reporting of Results ...................................................................................... 5
Understanding the Ping and RPM MIBs ...................................................................................... 6
Configuring RPM Probes ............................................................................................................ 9
Considerations for Configuring Timestamps for RPM ............................................................... 12
  Timestamps on M Series and T Series Routing Platforms ....................................................... 12
  Timestamps on J Series Services Routers ................................................................................ 14
  Timestamps on EX Series Ethernet Switches ........................................................................... 14
Configuring Probes Using SNMP ............................................................................................... 16
  Considerations for Destination Interface Configuration ....................................................... 16
Viewing Probe Results ................................................................................................................ 17
Configuring and Monitoring Real-Time Performance Using the J-Web User Interface ............... 24
Description and Deployment Scenario ......................................................................................... 26
  Scenario 1 ................................................................................................................................. 26
  Scenario 2 ................................................................................................................................. 29
Supported Routing and Switching Platforms ............................................................................ 32
  Hardware ................................................................................................................................. 32
Summary ....................................................................................................................................... 33
About Juniper Networks .............................................................................................................. 33

Table of Figures

Figure 1: RPM client-server operation ......................................................................................... 2
Figure 2: RPM hardware timestamping process ......................................................................... 3
Figure 3: Format of RPM timestamp .......................................................................................... 4
Figure 4: Probes per traffic forwarding class ............................................................................. 5
Figure 5: MIB OIDs for probes in each traffic class ................................................................... 8
Figure 6: Probe results from third-party MIB browser ............................................................. 9
Figure 7: Probe results from J-Web ............................................................................................ 24
Figure 8: RPM event viewer on J-Web ....................................................................................... 25
Introduction: Understanding Real-Time Performance Monitoring

This document provides an overview of real-time performance monitoring (RPM) features in Juniper Networks® JUNOS® Software, and provides information about tools that you can use for real-time performance monitoring on devices running JUNOS.

Real-time performance monitoring enables you to monitor network performance in real time and to assess and analyze network efficiency. Typically, network performance is assessed in real time based on the jitter, delay, and packet loss experienced on the network.

RPM is a service available in JUNOS that enables a router to measure metrics such as round-trip delays and unanswered echo requests. To achieve this, RPM exchanges a set of probes with other IP hosts in the network for monitoring and network tracking purposes. These probes are sent from a source node to other destination devices in the network that require tracking. Data such as transit delay and jitter can be collected from these probes, and this data can be used to provide an approximation of the delay and jitter experienced by live traffic in the network. Different live traffic metrics like round-trip time (RTT), positive egress jitter, negative egress jitter, positive ingress jitter, negative ingress jitter, positive round-trip jitter, and negative round-trip jitter can be gleaned from the results. RPM calculates minimum, maximum, average, peak-to-peak, standard deviation, and sum calculations for each of these measurements. RPM probes can also be used to verify the path between BGP neighbors.

Scope

This application note:

- Discusses the key concepts of RPM-based network probing for service-level agreement (SLA) measurements
- Explains the individual components of RPM probes
- Discusses the manageability aspect of RPM
- Provides sample configurations and output

JUNOS Software supports the following MIBs and tools that enable you to monitor delay and jitter on Juniper Networks devices:

- Standard DISMAN Ping MIB–RFC 2925 (standard on all Juniper Networks implementations) for creating probes
- Juniper Ping MIB for timestamp settings and threshold settings on probes
- Juniper enterprise-specific RPM MIB for gathering probe results and deriving Key Performance Indicators (KPIs)
- Command-line interface (CLI) commands
- SNMP set and get operations to configure probes and to gather information
- Juniper Networks J-Web Software user interface to monitor real-time performance

Design Considerations

RPM in its most simplistic form can be described as having a client (source) that sends out probe queries and a server (destination) that responds. During a probe, the client device sends a packet across to a remote server, which in turn returns the packet with an acknowledgement to the sender. Both the type and content of the queries sent from the client are user configurable. Both the source and destination nodes running RPM services are aware that the packets in the probe are used to compute information such as RTT and jitter delay. For each probe, RPM makes several measurements (for RTT as well as for different positive and negative jitter). For each type of measurement, RPM calculates the minimum, maximum, average, peak-to-peak, standard deviation, and overall sum over several different collections of measurements. For example, some of these collections include the current test, the most recently completed test, a “moving average” of n number of most recent probes, as well as all tests performed (including the one in progress).

Figure 1 shows a probe packet sent from the client and a corresponding reply from the server. The probe query and responses occur between specific user-defined source and destination addresses. The RPM service is a JUNOS process (rmopd) that runs on the Routing Engine.
The different packet types that can be used within the probe include:

- Internet Control Message Protocol (ICMP) echo
- ICMP timestamp
- HTTP get (not available for BGP RPM services)
- UDP echo
- TCP connection
- UDP timestamp

**Note:** `icmp-ping` is the default probe type on devices running JUNOS.

The probe packets are time stamped with the times at which they are sent and received at both the source and destination endpoints.

Table 1 lists the different timestamps that are discussed in the following Timestamps section, which also provides a detailed description of the processing of probe queries in the source and destination endpoints.

### Table 1: Types of Timestamps

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>DESCRIPTION</th>
<th>DIRECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Source (Client) Query (Request) Out</td>
<td>Egress</td>
</tr>
<tr>
<td>T2</td>
<td>Destination (Server) Query (Request) In</td>
<td>Ingress</td>
</tr>
<tr>
<td>T3</td>
<td>Destination (Server) Reply (Response) Out</td>
<td>Egress</td>
</tr>
<tr>
<td>T4</td>
<td>Source (Client) Reply (Response) In</td>
<td>Ingress</td>
</tr>
</tbody>
</table>

**Note:** In the case of ICMP probes, one of the deployment scenarios involves using RPM probes with a non-Juniper router as a server. In this case, the non-Juniper router responds to the RPM ICMP probes without adding the server timestamps and only the RTT delay is computed.

The main elements that are required for the RPM service to function correctly include:

- Timestamps
- RPM structure
- Configuration and reporting of results
- Destination interface

### Timestamps

Timestamping is needed to account for any latency in packet communications. Timestamps can be applied either on the client or server or on both client and server. All nodes in the network must be synchronized to a common, accurate clock source (preferably a Stratum 3 level) using Network Time Protocol (NTP). The probe types that support timestamping functionality include:

- icmp-ping
- icmp-ping-timestamp
- udp-ping
- udp-ping-timestamp
The timestamping activity consists of the source (client) node applying a timestamp (T1) to the RPM packets with the time at which they leave the node. The destination (server) node applies a timestamp (T2) when it receives the probe and a timestamp (T3) when the probe leaves the destination back to the source. The source receives the response and applies a timestamp (T4). Different metrics are calculated based on these timestamps collected from a series of probes. For example, the RTT can be computed as the difference between the timestamps T1 and T4 less T3 minus T2 (see Figure 2).

Timestamping can be performed using either hardware or software. The initial type of software or classical timestamping was done on the requester Routing Engine by the `rmopd` process when sending and receiving RPM probes. However, using the general OS affected the accuracy of the measurements, as the variability factor introduced by the general OS doing processing proved to be significantly larger than the amount of time spent by the packet on the wire.

The next type of timestamping was achieved using hardware and involved inserting T1-T4 timestamps in the RPM payload—a services PIC on Juniper Networks M Series Multiservice Edge Routers and Juniper Networks T Series Core Routers, and the `fwdd` process that spawns a real-time thread to do the timestamping on the Routing Engine of Juniper Networks J Series Services Routers in real time. (The `fwdd` process is used because there are no specific services PICs for the J Series).

Figure 2 illustrates the steps involved in the hardware timestamping process.

Using the one-way-hardware-timestamp allows the timestamp to be calculated in each direction as (T1-T2) or (T3-T4). One-way timestamps are valid only with hardware timestamps. Configuring the one-way-hardware-timestamp knob can enable this functionality to provide one-way measurements such as ingress delay, egress delay, ingress jitter, and egress jitter. Each one-way delay measurement is sanity checked against the corresponding round-trip measurement. If the value of the one-way delay measurement is greater than the round-trip time, then that sample is ignored.

RPM also can be configured without any hardware timestamps. Note: The breadth of measurements provided by RPM will be reduced if there are no server hardware timestamps. RPM also can be configured without any hardware timestamps. In this case, all of the timestamps are applied in software, which reduces the number as well as the accuracy of measurements even further. This application note is based on the testing done with hardware timestamps enabled.

One-way-hardware-timestamps are sensitive to inaccuracies in either server or client side clocks. So the clocks on the RPM server and client side need to be synchronized. Also, please note that there is no “two-way-hardware-timestamp” knob.
Figure 3 provides a detailed description of an RPM timestamp format. Each timestamp represents the time of day and consists of 4 bytes of seconds and 4 bytes of usecs in network order.

![Figure 3: Format of RPM timestamp](image)

**Understanding Probes and Tests**

Typically, a homogeneous set of probes, collectively called a test, is configured for each device on the network, and the information collected from these tests is stored in the Ping MIBs and the RPM MIB. Note that you cannot have multiple types of probes as part of a single test.

The probed target is specified using its IPv4 address, and each probed target is monitored over the course of a test that contains multiple probes. During a test, probes are generated and responses collected at a rate defined by the probe interval. A probe is considered lost if the probe interval expires before a response is received.

The probe owner and test names jointly identify each RPM configuration instance. A test consists of a range of probes over which the performance metrics are calculated.

**Note:** The term "test" has two different contexts within RPM. The first refers to the configuration name where it is combined with an owner name to identify a configuration instance. In the second case, it denotes a set of sent probes.

Probes can be one of the following types listed in Table 2. An RPM measurement can consist of multiple tests, each consisting of a different probe type and exchanged between a particular source-destination IP address pair. The interval between the probes and the tests are user configurable, as is the content of the probe’s data portion. The user can also control the number of probes that belong to a test. An algorithm within the software process determines the number of probes that are sent simultaneously from a source to a destination. Each probe can be assigned to a traffic forwarding class by specifying the type of DiffServ code point (DSCP) marking to be used in the IP packet header. The probe packets are treated based on the traffic class to which they belong when used in conjunction with logical tunnel (lt) or sp interface as the destination interface. Acceptable threshold limits of the different delays and latency can be set, and a failure or alarm can be triggered each time this threshold is exceeded.

**Note:** SNMP traps must be enabled for alarms to be triggered.

**Table 2: Probe Types and Function**

<table>
<thead>
<tr>
<th>PROBE TYPE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>udp-ping-timestamp</td>
<td>UDP timestamp requests sent to target address with T1,T2,T3,T4 timestamps</td>
</tr>
<tr>
<td>udp-ping</td>
<td>UDP packets sent to target. Supports complete range of hardware timestamps (T1-T4)</td>
</tr>
<tr>
<td>tcp-ping</td>
<td>TCP packets sent to target. Hardware timestamps are not supported</td>
</tr>
<tr>
<td>icmp-ping</td>
<td>ICMP echo requests sent to target address. Supports complete range of hardware timestamps (T1-T4)</td>
</tr>
<tr>
<td>icmp-ping-timestamp</td>
<td>ICMP timestamp requests sent to target address with T1,T2,T3,T4 timestamps</td>
</tr>
<tr>
<td>http-get (not available for BGP RPM)</td>
<td>HTTP get requests sent to target URL</td>
</tr>
<tr>
<td>http-metadata-get</td>
<td>HTTP get request for metadata to target URL</td>
</tr>
</tbody>
</table>
Figure 4 shows multiple probes, each belonging to a different traffic forwarding class sent from a Juniper Networks J6350 Services Router SDP source to a Session Description Protocol (SDP) destination. The two SDP routers connect over a service provider network of provider edge (PE) and provider (P) routers. The timestamping process is performed at the source and destination ends as described earlier.

All probes are considered to be unidirectional because a response is always expected and the remote destination node is considered to be down if the response to a request is not received. Thus, a single probe can be composed of a request and a corresponding response. Bidirectional probes can be considered as a requestor/responder pair that exists on each side of the connection. For example, a pair of routers acts as a requestor sending requests to the remote end while also functioning as responders to the requests coming in from the remote end.

Configuration and Reporting of Results

RPM probes can be configured from the JUNOS CLI, SNMP, and J-Web. The results can be reported via CLI or J-Web or SNMP. Alternately, a third-party MIB browser or performance management application can be used to retrieve RPM measurement data and plot the resulting graphs.

The following summarizes management interfaces available for RPM

<table>
<thead>
<tr>
<th>MANAGEMENT INTERFACE</th>
<th>INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLI</td>
<td>Command-line access to all configuration and monitoring of RPM</td>
</tr>
<tr>
<td>XML –JUNOScript</td>
<td>XML support for all CLI commands and capabilities</td>
</tr>
<tr>
<td>RFC 2925a</td>
<td>Standards-based IETF DISMAN-PING-MIB</td>
</tr>
<tr>
<td>MIB-JNX-RPM</td>
<td>• Supported as of JUNOS 9.6</td>
</tr>
<tr>
<td></td>
<td>• Juniper RPM MIB</td>
</tr>
<tr>
<td></td>
<td>• Leverages RFC 2925a</td>
</tr>
<tr>
<td></td>
<td>• Provides objects with the detailed RPM test results</td>
</tr>
<tr>
<td>MIB-JNX-PING</td>
<td>• Juniper Ping MIB</td>
</tr>
<tr>
<td></td>
<td>• Leverages RFC 2925a</td>
</tr>
<tr>
<td></td>
<td>• Adds objects relating to the different types of RPM probes</td>
</tr>
<tr>
<td>System log messages</td>
<td>Generated when configured thresholds are exceeded</td>
</tr>
<tr>
<td>SNMP traps</td>
<td>Generated when configured thresholds are exceeded</td>
</tr>
<tr>
<td>J-Web</td>
<td>GUI for configuration device to allow configuration and monitoring on individual device</td>
</tr>
</tbody>
</table>

For SNMP, MIB definitions allow SNMP operations on RPM variables. Two MIBs can be used for this purpose—the Ping MIB that is based on the RFC2925 [DISMAN-PING-MIB] Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations, and the RPM MIB or jnx-rpm.mib. These two MIBs can be used either together or separately to collect RPM data (although the jnx-rpm.mib provides more information because it is RPM-specific). The RPM MIB has been developed from the flat Ping MIB to provide a hierarchical collection of data.
The jnx-ping.mib allows creation of hardware timestamp-based probes, and it provides thresholds for raising alarms via SNMP to configure the number of samples in the moving average collection. Similar to jnx-ping.mib, the jnx-rpm.mib tables can be classified into a Results and History group respectively. However, in case of the jnx-rpm.mib, the data is classified into separate collection types and measurement sets.

**Note:** There is no additional license required to enable RPM services.

**Note:** The only option not configurable via SNMP is two-way hardware timestamping on the probe interfaces. See details about configuration samples in the Timestamp section.

The following is a list of RPM probe results that can be reported:

- Minimum, maximum, and average round-trip time as measured over the course of the test
- Standard deviation of the round-trip time measured over the course of the test
- Maximum jitter of the round trip time (difference between minimum and maximum RTT)
- Jitter of successive tests within a probe
- One way measurements for probes that have all four hardware timestamps present
- Number of probes sent during the test
- Number of probe responses received during the test
- Percentage of lost probes during the test

You can configure JUNOS Software to create system log entries and to generate SNMP traps when the threshold values configured for the parameters are exceeded. For more information about real-time performance monitoring traps, see “Real-Time Performance Monitoring Thresholds and Traps” later in this application note.

### Understanding the Ping and RPM MIBs

JUNOS supports the standard Ping MIB as defined in RFC 2925, as well as an enterprise-specific extension of the standard Ping MIB, jnx-ping.mib.

A ping test is used to determine whether packets sent from the local host reach the designated host and are returned. If the designated host can be reached, the ping test provides the approximate round-trip time. Ping test results are stored in the pingResultsTable and the pingProbeHistoryTable of the standard Ping MIB.

The enterprise-specific Ping MIB extends the standard Ping MIB control table. The enterprise-specific Ping MIB, whose object identifier is {jnxMIbs 7}, enables you to monitor network delay (latency), packet loss, network delay variation (jitter), one-way latency, and other network statistics. Items in this MIB are created when entries are created in the pingCtlTable of the standard Ping MIB. Each item is indexed exactly as in the standard Ping MIB.

For a downloadable version of the enterprise-specific extensions to the Ping MIB, see www.juniper.net/techpubs/software/junos/junos94/swconfig-net-mgmt/mib-jnx-ping.txt.

For a detailed interpretation of the enterprise-specific Ping MIB, see http://www.juniper.net/techpubs/software/junos/junos93/swconfig-net-mgmt/interpreting-the-enterprise-specific-pingmib.html.

The enterprise-specific RPM MIB enables you to access real-time performance-related data over SNMP. The RPM MIB represents a restructuring of the standard Ping MIB and converts the flat structure of the Ping MIB into a hierarchical collection of data. Similar to the Ping MIB, the RPM MIB also has two groups of tables: the Results group and the History group. The RPM MIB, however, groups its data into separate collection types such as currentTest and lastCompletedTest, and into measurement types such as roundTripTime and rttJitter. For a downloadable version of this MIB, see www.juniper.net/techpubs/software/junos/junos94/swconfig-net-mgmt/mib-jnx-rpm.txt.


Table 4 summarizes the MIBs involved for configuring and collecting the RPM probes and results. Later sections give detail and sample usage for these MIBs. The indexing for most of the tables in jnx-ping and jnx-rpm mibs are the same as standard RFC 2925 DISMAN-PING-MIB.

---

2If hardware timestamps are not available, RPM tries to make one-way measurements using sw timestamps in the icmp-ping-timestamp and the udpping-timestamp.
<table>
<thead>
<tr>
<th>MIB TABLE</th>
<th>MIB</th>
<th>USAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>pingCtlTable</td>
<td>DISMAN-PING-MIB (RFC 2925)</td>
<td>Table used to create a basic RPM probe.</td>
</tr>
<tr>
<td>pingResultsTable</td>
<td>DISMAN-PING-MIB (RFC 2925)</td>
<td>Table used for collection of ping test results like Min RTT and Max RTT of the latest test.</td>
</tr>
<tr>
<td>pingProbeHistoryTable</td>
<td>DISMAN-PING-MIB (RFC 2925)</td>
<td>Defines a table for storing the previous results of a ping operation.</td>
</tr>
<tr>
<td>jnxPingCtlTable</td>
<td>jnx-ping</td>
<td>Table used to set additional probe types like HTTP, VPN routing and forwarding [VRF] instance timestamps, and thresholds for RPM probes.</td>
</tr>
<tr>
<td>jnxPingResultsTable</td>
<td>jnx-ping</td>
<td>Augments the pingResultsTable with additional results.</td>
</tr>
<tr>
<td>jnxPingHistoryTable</td>
<td>jnx-ping</td>
<td>Augments the pingHistoryTable with additional data.</td>
</tr>
<tr>
<td>jnxPingLastTestResultTable</td>
<td>jnx-ping</td>
<td>Contains the results of the last completed ping tests.</td>
</tr>
<tr>
<td>jnxRpmResultsSampleTable</td>
<td>jnx-rpm</td>
<td>Provides measurements from the latest individual RPM probe samples. This is the table which contains the results for RTT, jitter, egress jitter, and so on for the latest completed test.</td>
</tr>
<tr>
<td>jnxRpmResultsSummaryTable</td>
<td>jnx-rpm</td>
<td>Provides a summary of the results for each RPM entry and for each data collection maintained by that entry. The RPM feature maintains several different collections of probe data, providing overall summaries as well as detailed calculations for each collection.</td>
</tr>
<tr>
<td>jnxRpmResultsCalculatedTable</td>
<td>jnx-rpm</td>
<td>Provides a set of calculated values for each RPM entry, for each collection of probes maintained within that entry, and for each supported measurement set within that collection of probes.</td>
</tr>
<tr>
<td>jnxRpmHistorySampleTable</td>
<td>jnx-rpm</td>
<td>Provides measurements for each sample stored in the history table of RPM probe entries. In addition to the last completed probe, this table also provides data for a configurable number of most recent probes [all history tables in this MIB provide the same number of entries as the pingProbeHistoryTable]. However, the table does not maintain entries for: unsuccessful probes and invalid measurement types. This table along with jnxRpmResultsSampleTable can be used to derive RTT and jitter over a period of time for graphs.</td>
</tr>
<tr>
<td>jnxRpmHistorySummaryTable</td>
<td>jnx-rpm</td>
<td>Similar to the jnxRpmResultsSummaryTable, this table provides summary data for each collection of probes within each RPM entry. In addition to summary data for the current probe, the table also provides summary information for a number of the most recent probes. You can configure the number of most recent probes that should be stored in this table.</td>
</tr>
<tr>
<td>jnxRpmHistoryCalculatedTable</td>
<td>jnx-rpm</td>
<td>As with the jnxRpmResultsCalculatedTable, this table provides a set of calculated values for each RPM entry, for each collection of probes maintained within that entry, and for each supported calculated type within that collection of probes.</td>
</tr>
</tbody>
</table>

**Note:** The jnxRpmResultsSampleTable returns the most recently completed samples. The jnxRpmHistorySampleTable gives the results of the number \( n \) of most recent samples.
Figure 5 shows the MIB object identifiers (OIDs) for each of the probes that are configured for different traffic forwarding classes. The different metrics such as RTT, jitter, and so on are tracked for each of the probes belonging to a different traffic class.

Figure 5: MIB OIDs for probes in each traffic class
Figure 6 shows the metrics obtained from a third-party MIB browser.

![Figure 6: Probe results from third-party MIB browser](image)

**Configuring RPM Probes**

You can configure probes and probe parameters by including the probe statement and its attributes at the `[edit services rpm]` hierarchy level using JUNOS CLI as below.

```cli
owner owner {
    test test-name {
        data-fill data;
        data-size size;
        destination-interface interface-name;
        destination-port port;
        dscp-code-point dscp-bits;
        hardware-timestamp;
        history-size size;
        moving-average-size number;
        one-way-hardware-timestamp;
        probe-count count;
        probe-interval seconds;
        probe-type type;
        routing-instance instance-name;
        source-address address;
        target (url url | address address);
        test-interval interval;
        thresholds thresholds;
        traps traps;
    }
}
```

For more information about configuration statements at the `[show services rpm probe]` hierarchy, see the *JUNOS Services Interfaces Configuration Guide*. 

---

**Figure 6**
Probes results from third-party MIB browser
Table 5 shows some of the most common options that you can configure for real-time performance monitoring tests using JUNOS CLI and corresponding SNMP parameters.

**Note:** You can use either the JUNOS CLI configuration or the J-Web graphical user interface (GUI) or SNMP to view and configure the fields listed in Table 5. The following sections contain more information about using the JUNOS CLI and J-Web GUI for configuring and monitoring RPM.

**Table 5: Real-Time Performance Monitoring Configuration Options for CLI and SNMP**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>CLI CONFIGURATION STATEMENT</th>
<th>CORRESPONDING SNMP PARAMETERS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probe owner</td>
<td>Probe</td>
<td>pingCtlOwnerIndex and jnxPingCtlOwnerIndex</td>
<td>Identifies the probe owner and enables logical grouping of tests. The probe owner identifier can be up to 32 characters long.</td>
</tr>
<tr>
<td>Test name</td>
<td>Test</td>
<td>pingCtlTestName and jnxPingCtlTestName</td>
<td>Specifies the test name. The test name can be up to 32 characters long.</td>
</tr>
</tbody>
</table>
| Probe type  | probe-type                  | pingCtlType                   | Specifies the type of probe to be sent as part of the test. The available probe types are:  
  • http-get  
  • http-get-metadata  
  • icmp-ping  
  • icmp-ping-timestamp  
  • tcp-ping  
  • udp-ping |
| Target address | target-address             | pingCtlTargetAddressType and pingCtlTargetAddress | Sets the destination of this probe. |
| Probe interval | probe-interval             | pingCtlTimeOut                | Sets the wait time [in seconds] between each probe transmission. The range is 1 to 255 seconds. |
| Test interval | test-interval               | pingCtlFrequency              | Sets the wait time [in seconds] between tests. The range is 0 to 86,400 seconds. |
| Source address | source-address             | pingCtlSourceAddressType and pingCtlSourceAddress | Source address of probe (if this is not specified, the probe uses the outgoing interface address as the source). |
| Thresholds  | Thresholds                  | See SNMP parameters in Real-Time Performance MonitoringThresholds and Traps section below | Assigns test and probe limits. Please see section on Thresholds and Traps below for details of configuration under this hierarchy. |
| Traps       | Traps                       | jnxPingCtlTrapGeneration (See details in later sections) | Generates SNMP traps when specified thresholds are exceeded. Please see section on Thresholds and Traps below for details. |
| Probe count | probe-count                 | pingCtlProbeCount             | Specifies the total number of probes sent for each test. The range is 1 to 15 probes. |
Table 5: Real-Time Performance Monitoring Configuration Options for CLI and SNMP (continued)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>CLI CONFIGURATION STATEMENT</th>
<th>CORRESPONDING SNMP PARAMETERS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination port</td>
<td>destination-port&lt;br&gt;Hierarchy level: [edit services rpm probe owner test test-name]</td>
<td>jnxPingCtlTargetPort</td>
<td>Specifies the TCP or UDP port to which the probes are sent. You can either use port number 7 (a standard TCP or UDP port number) or select a port number from 49,152 through 65,535.</td>
</tr>
<tr>
<td>DSCP bits</td>
<td>dscp-code-points&lt;br&gt;Hierarchy level: [edit services rpm probe owner test test-name]</td>
<td>pingCtlDSField</td>
<td>Sets the DSCP bits corresponding to traffic class in the IP header. The value must be a valid 6-bit pattern. The default is 000000.</td>
</tr>
<tr>
<td>Data size</td>
<td>data-size&lt;br&gt;Hierarchy level: [edit services rpm probe owner test test-name]</td>
<td>pingCtlDataSize</td>
<td>Specifies the size (in bytes) of the data portion of the ICMP probes. The range is 0 to 65,507 bytes.</td>
</tr>
<tr>
<td>Data fill</td>
<td>data-fill&lt;br&gt;Hierarchy level: [edit services rpm probe owner test test-name]</td>
<td>pingCtlDataFill</td>
<td>Specifies the contents of the data portion of the ICMP probes; must be a hexadecimal value in the range of 1 to 800h.</td>
</tr>
<tr>
<td>Destination interface</td>
<td>destination-interface&lt;br&gt;Hierarchy level: [edit services rpm probe owner test test-name]</td>
<td>pingCtlIfIndex OR jnxPingCtlIfName AND pingCtlByPassRouteTable</td>
<td>Identifies an interface on the RPM client; enables hardware timestamping on M Series, T Series, and EX3200 and EX4200 Ethernet Switches.</td>
</tr>
<tr>
<td>Hardware timestamp</td>
<td>hardware-timestamp&lt;br&gt;Hierarchy level: [edit services rpm probe owner test test-name]</td>
<td>jnxPingCtlTargetPort</td>
<td>Enables hardware-based timestamping of RPM probe messages on router. Note: For the J Series only, the fwdd process does this since services hardware is “simulated.”</td>
</tr>
<tr>
<td>One-way timestamp</td>
<td>one-way-hardware-timestamp&lt;br&gt;Hierarchy level: [edit services rpm probe owner test test-name]</td>
<td>jnxPingCtlOneWayHWTimestamp</td>
<td>Uses all four timestamps in the packet so that it’s possible to measure one-way propagation characteristics to and from the probe target.</td>
</tr>
<tr>
<td>History</td>
<td>history-size&lt;br&gt;Hierarchy level: [edit services rpm probe owner test test-name]</td>
<td>pingCtlMaxRows</td>
<td>Number of stored history entries [0..255]</td>
</tr>
<tr>
<td>Moving average</td>
<td>moving-average-size&lt;br&gt;Hierarchy level: [edit services rpm probe owner test test-name]</td>
<td>jnxPingCtlMovAvgSize</td>
<td>Number of samples used for moving average [0..255]</td>
</tr>
<tr>
<td>Routing instance</td>
<td>routing-instance&lt;br&gt;Hierarchy level: [edit services rpm probe owner test test-name]</td>
<td>jnxPingCtlRoutingInstanceName</td>
<td>Routing instance used by probes for VRF ping</td>
</tr>
</tbody>
</table>

Note: There are other SNMP parameters for SNMP-specific triggering, such as pingCtlAdminStatus and pingCtlRowStatus, which are explained later in this application note.
Considerations for Configuring Timestamps for RPM

To account for latency in the communication of probe messages, you can enable timestamping of the probe packets. Timestamps are only supported for the following probe types:

- icmp-ping
- icmp-ping-timestamp
- udp-ping
- udp-ping-timestamp

The configuration is handled in different ways on Juniper Networks M Series Multiservice Edge Routers, Juniper Networks T Series Core Routers, Juniper Networks J Series Services Routers, and Juniper Networks EX Series Ethernet Switches as described in the following sections:

- Timestamps on M Series and T Series routing platforms
- Timestamps on J Series Services Routers
- Timestamps on Juniper Networks EX Series Ethernet Switches

Timestamps on M Series and T Series Routing Platforms

If your router network includes an Adaptive Services (AS) or MultiServices PIC, you can enable timestamping of RPM probe messages. The timestamp is applied on the RPM client router [the router that originates the RPM probes] and the RPM probe server [the router that responds to RPM probes] for IPv4 traffic only. It is supported in the Layer 2 service package on all MultiServices PICs and in the Layer 3 service package on AS and MultiServices PICs.

To configure two-way timestamping on M Series and T Series routing platforms, include the destination-interface statement at the [edit services rpm probe probe-owner test test-name] hierarchy level. For example:

```
[edit]
services rpm probe probe-owner test test-name {
    destination-interface sp-fpc/pic/port.logical-unit;
}
```

Specify the RPM client router on the logical interface on the AS PIC by including the rpm statement at the [edit interfaces interface-name unit logical-unit-number] hierarchy level. For example:

```
[edit]
interfaces interface-name unit logical-unit-number {
    rpm client;
}
```

**Note:** This cannot be set using SNMP.

The logical interface (unit 0) must be dedicated to the RPM task. It requires configuration of the family inet statement and a /32 address, as shown in the example given at the end of this section. This configuration is also needed for other services such as Network Address Translation (NAT) and stateful firewall.

**Note:** If you configure RPM timestamping on an AS PIC, you cannot configure the source-address statement at the [edit services rpm probe probe-name test test-name] hierarchy level. To configure one-way timestamping, you must also include the one-way-hardware-timestamp statement at the [edit services rpm probe probe-owner test test-name] hierarchy level. For example:

```
[edit]
services rpm probe probe-owner test test-name {
    one-way-hardware-timestamp;
}
```

**Note:** If you are using one-way timestamps, make sure that the clocks on both the sender and receiver are synchronized.
**Note:** If you configure RPM probes for a services interface (sp-), you need to announce local routes in a specific way for the following routing protocols:

- For OSPF, you can announce the local route by including the services interface in the OSPF area. To configure this setting, include the interface sp-fpc/pic/port statement at the [edit protocols ospf area area-number] hierarchy level.

- For BGP and IS-IS, you must export interface routes and create a policy that accepts the services interface local route. To export interface routes, include the point-to-point and lan statements at the [edit routing-options interface-routes family inet export] hierarchy level. To configure an export policy that accepts the services interface local route, include the protocol local, rib inet.0, and route-filter sp-interface-ip-address/32 exact statements at the [edit policy-options policy-statement policy-name term term-name from] hierarchy level, and the accept action at the [edit policy-options policy-statement policy-name term term-name then] hierarchy level. For the export policy to take effect, apply the policy to BGP or IS-IS with the export policy-name statement at the [edit protocols protocol-name] hierarchy level.


Routing the probe packets through the AS or MultiServices PIC also enables you to filter the probe packets to particular queues. The following example shows the RPM configuration and the filter that specifies queuing:

**RPM Client Configuration**

```
services rpm {
    probe p1 {
        test t1 {
            probe-type icmp-ping;
            target address 10.8.4.1;
            probe-count 10; probe-interval 10;
            test-interval 10;
            dscp-code-points af11;
            data-size 100;
            destination-interface sp-1/2/0.0;
        }
    }
}
firewall {
    filter f1 {                    #defines the filter
        term t1 {
            from {
                dscp af11;
            }
            then {
                forwarding-class assured-forwarding;
            }
        }
    }
}
interfaces sp-1/2/0 {
    unit 0 {
        rpm client;
        family inet {
            address 10.8.4.2/32;
            filter {
                input f1;
            }
        }
    }
}
```
Server Configuration: Server configuration cannot be done using SNMP

interfaces sp-1/2/0 {
    unit 0 {
        rpm server;
        family inet {
            address 10.8.4.1/32;
        }
    }
}

For more information about firewall filters, see the JUNOS Policy Framework Configuration Guide. For more information about queuing, see the JUNOS Class of Service Configuration Guide.

Timestamps on J Series Services Routers
On J Series Services Routers, timestamps are applied in the forwarding process (fwdd-rt).

- To configure two-way timestamping, include the hardware-timestamp statement at the [edit services rpm probe probe-owner test test-name] hierarchy level:

  [edit]
services rpm probe probe-owner test test-name {
    hardware-timestamp;
}

- To configure one-way timestamping, include both the hardware-timestamp and the one-way-hardware-timestamp statements at the [edit services rpm probe probe-owner test test-name] hierarchy level:

  [edit]
services rpm probe probe-owner test test-name {
    hardware-timestamp;
    one-way-hardware-timestamp;
}

For more information, see the J Series Services Router Administration Guide.

Note: You cannot include the source-address statement at the [edit services rpm probe probe-name test test-name] hierarchy level if you have configured the hardware-timestamp or one-way-hardware-timestamp statement.

Timestamps on EX Series Ethernet Switches
To configure two-way timestamping on EX Series switches, include the destination-interface statement at the [edit services rpm probe probe-owner test test-name] hierarchy level:

- destination-interface ge-fpc/pic/port.logical-unit

Specify the RPM client router on the ge logical interface by including the rpm client statement at the [edit interfaces interface-name unit logical-unit-number] hierarchy level:

- rpm client

Specify the RPM server router on the ge logical interface by including the rpm server statement at the [edit interfaces interface-name unit logical-unit-number] hierarchy level:

- rpm server

Note: This cannot be set using SNMP
Example
Client configuration
EX3200# show services rpm
probe mad {
    test mad-udp-ping {
        probe-type udp-ping;
        target address 20.20.20.2;
        probe-count 5;
        test-interval 20;
        destination-port 49161;
        dscp-code-points af22;
        destination-interface ge-0/0/15.0;
    }
    test mad-ping {
        probe-type icmp-ping;
        target address 20.20.20.2;
        probe-count 10;
        test-interval 30;
        dscp-code-points af43;
        destination-interface ge-0/0/15.0;
    }
}
probe-limit 500;

EX3200# show interfaces ge-0/0/15
unit 0 {
    rpm {
        client;***
    }
    family inet {
        address 20.20.20.1/24;
    }
}

***:Interface RPM client and server cannot be configured using SNMP.

EX3200> show configuration services rpm
probe-server {
    udp {
        port 49161;
    }
}

EX3200# show interfaces ge-0/0/15
unit 0 {
    rpm {
        server;
    }
    family inet {
        address 20.20.20.2/24;
    }
}
Configuring Probes Using SNMP

To configure a probe using SNMP, follow these steps:

1. Set the createAndWait (5) value for pingCtlRowStatus.

   ```
   snmpset -v2c -c private 172.22.54.175 pingCtlRowStatus.4.66.68.49.1.65 i 5
   ```

2. Set the pingCtlTable and jnxpingCtlTable objects.

   ```
   snmpset -v2c -c private 172.22.54.175
   pingCtlTargetAddress.4.66.68.49.1.65 h 0xac1636ac p
   pingCtlTargetAddressType.4.66.68.49.1.65 i 1
   pingCtlSourceAddress.4.66.68.49.1.65 h 0x0ace0001 p
   pingCtlSourceAddressType.4.66.68.49.1.65 i 1
   pingCtlAdminStatus.4.66.68.49.1.65 i 2
   pingCtlDataSize.4.66.68.49.1.65 u 64
   pingCtlProbeCount.4.66.68.49.1.65 u 10
   pingCtlFrequency.4.66.68.49.1.65 u 300
   pingCtlType.4.66.68.49.1.65 o jnxPingUdpTimestamp
   pingCtlDSField.4.66.68.49.1.65 u 104
   pingCtlTimeOut.4.66.68.49.1.65 u 2
   jnxPingCtlTargetPort.4.66.68.49.1.65 u 64001
   ```

3. Set the value of pingCtlRowStatus to active (1).

   ```
   snmpset -v2c -c private 172.22.54.175 pingCtlRowStatus.4.66.68.49.1.65 i 1
   ```

4. pingCtlAdminStatus to enable(1) and:

   ```
   snmpset -v2c -c private 172.22.54.175 pingCtlAdminStatus.4.66.68.49.1.65 i 1
   ```

**Note:** RPM configurations set using SNMP are lost during device restart.

**Note:** We recommend that you do not configure more than 50 probes on a device.

Considerations for Destination Interface Configuration

A destination interface can be either sp interface on the M Series/T Series routers, or lt interface on the J Series routers, or ge interfaces on EX Series switches. In case of the M Series and T Series routers, the destination sp interface is used for hardware timestamping. The lt interface on the J Series router supports the classification of the probes based on user defined classifiers.

**Note:** Because this document is based on tests executed on a J Series router, only usage of the lt interface is being discussed here in detail.

A Logical Tunnel (lt) interface must be defined as the destination interface on the remote node when RPM services with DSCP marking-based class of service (CoS) are enabled on a J Series router. RPM service requires CoS to measure SLA. As mentioned earlier, RPM probe packets are generated by the Routing Engine and are classified as Network Control (NC) packets by default. Hence, any user-configured DSCP marking that would involve classification into other queues is not applied to the probe packets. The lt interface provides a workaround for this condition. It spins the probe packets through the Packet Forwarding Engine (PFE) so that the probe packets resemble transit traffic that is generated external to the router. After this, the packets can be classified into the appropriate CoS queues. As a result, the lt interface on the J Series router is required for CoS to be applied on the probe packets based on DSCP marking. The lt interface must be defined as the destination interface when extending CoS support on the J Series router. Pertaining to the M Series and T Series, the sp interface on the services PIC is used as the destination interface.

In scenarios where DSCP marking is required on the RPM probe packets, the CoS functionality (traffic classes, shaping/scheduling, rewrite rules, etc.) must be explicitly defined. The lt interface must be configured under the CoS interfaces section. Table 6 shows a snippet of the CoS-based information for the lt interface.
### Table 6: Example of CoS Interface CLI Output

<table>
<thead>
<tr>
<th>Logical interface: lt-0/0/0.0, Index: 72</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Classifier</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logical interface: lt-0/0/0.1, Index: 73</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Classifier</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logical interface: lt-0/0/0.10, Index: 74</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Classifier</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logical interface: lt-0/0/0.20, Index: 75</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Classifier</strong></td>
</tr>
</tbody>
</table>

### Viewing Probe Results

The `show services rpm probe-results` command returns the following output:

```
root@device> show services rpm probe-results
Owner: p1, Test: t1
Target address: 10.8.4.1, Probe type: icmp-ping
  Destination interface name: sp-1/2/0.0
  Test size: 10 probes
  Probe results:
  Response received, Thu Nov 30 11:07:38 2006
  Rtt: 497 usec
  Results over current test:
    Probes sent: 10, Probes received: 10, Loss percentage: 0
    Measurement: Round trip time
    Samples: 10, Minimum: 465 usec, Maximum: 497 usec, Average: 477 usec,
    Peak to peak: 32 usec, Stddev: 9 usec
  Results over last test:
    Probes sent: 10, Probes received: 10, Loss percentage: 0
    Test completed on Thu Nov 30 11:07:38 2006
    Measurement: Round trip time
    Samples: 10, Minimum: 465 usec, Maximum: 497 usec, Average: 477 usec,
    Peak to peak: 32 usec, Stddev: 9 usec
  Results over all tests:
    Probes sent: 30, Probes received: 30, Loss percentage: 0
    Measurement: Round trip time
    Samples: 30, Minimum: 465 usec, Maximum: 30332 usec, Average: 1857 usec,
    Peak to peak: 29867 usec, Stddev: 5550 usec
```
You can also poll the following MIB tables to obtain this information through SNMP. See the Description and Deployment Scenario section for sample outputs from some of these tables:

- jnxRpmResultsSampleTable (defined in jnx-rpm.mib)
- jnxRpmHistorySampleTable (defined in jnx-rpm.mib)
- jnxRpmResultsSummaryTable (defined in jnx-rpm.mib)
- jnxRpmHistorySummaryTable (defined in jnx-rpm.mib)
- jnxRpmResultsCalculatedTable (defined in jnx-rpm.mib)
- jnxRpmHistoryCalculatedTable (defined in jnx-rpm.mib)

Table 7 shows RPM Key performance Indicators and related MIBs. The jnx-rpm MIB is a superset of Ping MIB and therefore all RPM results need to be monitored using jnx-rpm MIB alone.

### Table 7:  RPM KPIs with related MIBs

<table>
<thead>
<tr>
<th>KPI</th>
<th>MIBS</th>
<th>HISTORY MIB</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>jnxRpmResSampleValue</td>
<td>jnxRpmHistSampleValue</td>
<td>jnxRpmResSampleValue from jnx-RPM table with measurement type roundTripTime gives data for latency measurement.</td>
</tr>
<tr>
<td>Dropped packet count</td>
<td>jnxRpmResSumSent</td>
<td>jnxRpmHistSumSent</td>
<td>The RPM MIB table jnxRpmResultsSummaryTable gives the packet sent, received, and percentage loss for current, last, allTests, and for moving average (the number of most recent probes where n is configurable).</td>
</tr>
<tr>
<td></td>
<td>jnxRpmResSumReceived</td>
<td>jnxRpmHistSumReceived</td>
<td></td>
</tr>
<tr>
<td></td>
<td>jnxRpmResSumPercentLost</td>
<td>jnxRpmHistSumPercentLost</td>
<td></td>
</tr>
<tr>
<td>Jitter (egress, ingress, RTT)</td>
<td>jnxRpmResSampleValue</td>
<td>jnxRpmHistSampleValue</td>
<td>For jitter, the RPM Table jnxRpmResultsSampleTable should be used to measure the following measurement types</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- rttJitter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- rttInterarrivalJitter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- egressJitter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- egressInterarrivalJitter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- ingressJitter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- ingressInterarrivalJitter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NOTE: If the latest probe is unsuccessful, no measurement types will be available. Not all measurements are available for all tests and probe types due to various checks and hardware timestamps required. Two-way-hardware-timestamp is required for most of the jitter measurements.</td>
</tr>
</tbody>
</table>
### Table 7: RPM KPIs with related MIBs (continued)

<table>
<thead>
<tr>
<th>KPI</th>
<th>MIBS</th>
<th>HISTORY MIB</th>
<th>DESCRIPTION</th>
</tr>
</thead>
</table>
| Delay (egress and ingress) | jnxRpmResSampleValue | jnxRpmHistSampleValue | For ingress and egress delays (only when one-way-hardware-timestamp is enabled), jnxRpmResultsSampleTable should be used with the following measurement types:  
  - Egress  
  - Ingress |

| RPM probe calculations | jnxRpmResCalcSamples | jnxRpmHistCalcSamples | The jnxRpmResultsCalculatedTable gives a set of calculated values for each RPM entry, for each collection of probes maintained within that entry, and for each supported measurement set within that collection of probes. For example, for measurement type RTT, this table gives number of samples considered for calculation, maximum value of RTT, minimum, average, and PktoPK similar to the one shown in CLI example below.  
  Measurement: RTT  
  Samples: 1, Minimum: 514 usec  
  Maximum: 514 usec, Average: 514 usec,  
  Peak to peak: 0 usec, Stddev: 0 usec, Sum: 514 usec |
Real-Time Performance Monitoring Thresholds and Traps

JUNOS Software enables you to configure threshold values for certain probe parameters so that network behavior can be measured against an acceptable value. Table 8 lists parameters for which you can configure threshold values.

Table 8: Real-Time Monitoring Thresholds

<table>
<thead>
<tr>
<th>PARAMETER/JWEB GUI FIELD</th>
<th>CONFIGURATION STATEMENT</th>
<th>SNMP</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Successive lost probes</td>
<td>successive-loss</td>
<td>pingCtlTrapProbeFailureFilter</td>
<td>Total number of probes that must be lost successively to trigger a probe failure and generate a system log message. The range is 0 to 15 probes.</td>
</tr>
<tr>
<td></td>
<td>Hierarchy level: [edit services rpm probe owner test test-name thresholds]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lost probes</td>
<td>total-loss</td>
<td>pingCtlTrapTestFailureFilter</td>
<td>Total number of probes that must be lost to trigger a probe failure and generate a system log message. The range is 0 to 15 probes.</td>
</tr>
<tr>
<td></td>
<td>Hierarchy level: [edit services rpm probe owner test-name thresholds]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Round-trip time</td>
<td>Rtt Hierarchy level:</td>
<td>jnxPingCtlRttThreshold</td>
<td>Total round-trip time [in microseconds] from the services router to the remote server, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.</td>
</tr>
<tr>
<td></td>
<td>[edit services rpm probe owner test-name thresholds]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jitter</td>
<td>jitter-rtt Hierarchy</td>
<td>jnxPingCtlRttJitterThreshold</td>
<td>Total jitter [in microseconds] for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.</td>
</tr>
<tr>
<td></td>
<td>level: [edit services rpm probe owner test-name thresholds]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard deviation</td>
<td>std-dev-rtt Hierarchy</td>
<td>jnxPingCtlRttStdDevThreshold</td>
<td>Maximum allowable standard deviation [in microseconds] for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.</td>
</tr>
<tr>
<td></td>
<td>level: [edit services rpm probe owner test-name thresholds]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Egress time</td>
<td>egress-time Hierarchy</td>
<td>jnxPingCtlEgressTimeThreshold</td>
<td>Total one-way time [in microseconds] from the router to the remote server, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.</td>
</tr>
<tr>
<td></td>
<td>level: [edit services rpm probe owner test-name thresholds]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ingress time</td>
<td>ingress-time Hierarchy</td>
<td>jnxPingCtlIngressTimeThreshold</td>
<td>Total one-way time [in microseconds] from the remote server to the router, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.</td>
</tr>
<tr>
<td></td>
<td>level: [edit services rpm probe owner test-name thresholds]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 8: Real-Time Monitoring Thresholds (continued)

<table>
<thead>
<tr>
<th>PARAMETER/JWEB GUI FIELD</th>
<th>CONFIGURATION STATEMENT</th>
<th>SNMP</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jitter egress time</td>
<td>jitter-egress Hierarchy level: [edit services rpm probe owner test test-name thresholds]</td>
<td>jnxPingCtlEgressJitterThreshold</td>
<td>Total outbound-time jitter (in microseconds) for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.</td>
</tr>
<tr>
<td>Jitter ingress time</td>
<td>jitter-ingress Hierarchy level: [edit services rpm probe owner test test-name thresholds]</td>
<td>jnxPingCtlIngressJitterThreshold</td>
<td>Total inbound-time jitter (in microseconds) for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.</td>
</tr>
<tr>
<td>Egress standard deviation</td>
<td>std-dev-egress Hierarchy level: [edit services rpm probe owner test test-name thresholds]</td>
<td>jnxPingCtlEgressStdDevThreshold</td>
<td>Maximum allowable standard deviation of outbound times (in microseconds) for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.</td>
</tr>
<tr>
<td>Ingress standard deviation</td>
<td>std-dev-ingress Hierarchy level: [edit services rpm probe owner test test-name thresholds]</td>
<td>jnxPingCtlIngressStdDevThreshold</td>
<td>Maximum allowable standard deviation of inbound times (in microseconds) for a test, which, if exceeded, triggers a probe failure and generates a system log message. The range is 0 to 60,000,000 microseconds.</td>
</tr>
</tbody>
</table>

You can configure JUNOS Software to generate the following traps and system log entries when the thresholds listed in Table 9 are exceeded:
### Table 9: Real-Time Performance Monitoring Traps and System Log Entries and Configuration for Traps

<table>
<thead>
<tr>
<th>Trap</th>
<th>TRAP Object ID</th>
<th>CLI Option to Set to Generate This Trap</th>
<th>SNMP Parameter to Set for Generating the Trap (Set in BITS)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pingProbeFailed</td>
<td>1.3.6.1.2.1.80.0.1</td>
<td>probe-failure</td>
<td>pingCtlTrapGeneration - probeFailure(0)</td>
<td>Generated every time the pingCtlTrapProbeFailureFilter number of consecutive probes fails during a test. SYSLOG tag for this is SNMP_TRAP_PING_PROBE_FAILED.</td>
</tr>
<tr>
<td>pingTestFailed</td>
<td>1.3.6.1.2.1.80.0.2</td>
<td>test-failure</td>
<td>pingCtlTrapGeneration - testFailure(1)</td>
<td>Generated when the test is completed and at least the pingCtlTrapTestFailureFilter number of probes fails.</td>
</tr>
<tr>
<td>pingTestCompleted</td>
<td>1.3.6.1.2.1.80.0.3</td>
<td>test-completion</td>
<td>pingCtlTrapGeneration - testCompletion(2)</td>
<td>Generated when the test is completed and fewer than the pingCtlTrapTestFailureFilter probes fail. SYSLOG tag for this is SNMP_TRAP_PING_TEST_COMPLETED.</td>
</tr>
<tr>
<td>jnxPingRttThresholdExceeded</td>
<td>1.3.6.1.4.1.2636.4.9.0.1</td>
<td>rtt-exceeded</td>
<td>jnxPingCtlTrapGeneration - rttThreshold(0)</td>
<td>Generated when the maximum round-trip time is exceeded by any probe.</td>
</tr>
<tr>
<td>jnxPingRttStdDevThresholdExceeded</td>
<td>1.3.6.1.4.1.2636.4.9.0.2</td>
<td>std-dev-exceeded</td>
<td>jnxPingCtlTrapGeneration - rttStdDevThreshold(1)</td>
<td>Generated when the calculated standard deviation of the round-trip time at the end of any test exceeds the maximum allowed value.</td>
</tr>
<tr>
<td>jnxPingRttJitterThresholdExceeded</td>
<td>1.3.6.1.4.1.2636.4.9.0.3</td>
<td>jitter-exceeded</td>
<td>jnxPingCtlTrapGeneration - rttJitterThreshold(2)</td>
<td>Generated when the jitter in the round-trip time over the course of any test exceeds the maximum allowed jitter.</td>
</tr>
<tr>
<td>jnxPingEgressThresholdExceeded</td>
<td>1.3.6.1.4.1.2636.4.9.0.4</td>
<td>egress-time-exceeded</td>
<td>jnxPingCtlTrapGeneration - egressThreshold(3)</td>
<td>Generated when a probe exceeds the maximum allowed egress trip time. This trap is supported only for jnxPingICMPTimeStamp probes.</td>
</tr>
</tbody>
</table>

The CLI options for generating traps are set using the `owner test test-name traps` command. The SNMP parameters are set in octal format and must be entered in the corresponding object's configuration.
### Table 9: Real-Time Performance Monitoring Traps and System Log Entries and Configuration for Traps (continued)

<table>
<thead>
<tr>
<th>Trap</th>
<th>TRAP Object ID</th>
<th>CLI Option to Set to Generate This Trap</th>
<th>SNMP Parameter to Set for Generating the Trap (Set in BITS)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jnxPingEgressStdDevThresholdExceeded</td>
<td>1.3.6.1.4.1.2636.4.9.0.5</td>
<td>egress-std-dev-exceeded</td>
<td>jnxPingCtlTrapGeneration - egressStdDevThreshold(4)</td>
<td>Generated if the calculated standard deviation of the egress trip time at the end of any test exceeds the maximum allowed limit. This trap is supported only for jnxPingIcmpTimeStamp probes.</td>
</tr>
<tr>
<td>jnxPingEgressJitterThresholdExceeded</td>
<td>1.3.6.1.4.1.2636.4.9.0.6</td>
<td>egress-jitter-exceeded</td>
<td>jnxPingCtlTrapGeneration - egressJitterThreshold(5)</td>
<td>Generated when the jitter in the egress trip time over the course of any test exceeds the maximum allowed value. This trap is supported only for jnxPingIcmpTimeStamp probes.</td>
</tr>
<tr>
<td>jnxPingIngressThresholdExceeded</td>
<td>1.3.6.1.4.1.2636.4.9.0.7</td>
<td>ingress-time-exceeded</td>
<td>jnxPingCtlTrapGeneration - ingressThreshold(6)</td>
<td>Generated when a probe exceeds the maximum allowed ingress trip time. This trap is supported only for jnxPingIcmpTimeStamp probes.</td>
</tr>
<tr>
<td>jnxPingIngressStddevThresholdExceeded</td>
<td>1.3.6.1.4.1.2636.4.9.0.8</td>
<td>ingress-std-dev-exceeded</td>
<td>jnxPingCtlTrapGeneration - ingressStdDevThreshold(7)</td>
<td>Generated if the calculated standard deviation of the ingress trip time at the end of any test exceeds the maximum allowed limit.</td>
</tr>
<tr>
<td>jnxPingIngressJitterThresholdExceeded</td>
<td>1.3.6.1.4.1.2636.4.9.0.9</td>
<td>ingress-time-exceeded</td>
<td>jnxPingCtlTrapGeneration - ingressJitterThreshold(8)</td>
<td>Generated when the jitter in the ingress trip time over the course of any test exceeds the maximum allowed value. This trap is supported only for jnxPingIcmpTimeStamp probes.</td>
</tr>
</tbody>
</table>
Configuring and Monitoring Real-Time Performance Using the J-Web User Interface

If you are using J-Web to manage your JUNOS devices, you can use the J-Web graphical user interface (GUI) to configure and monitor real-time performance probes.

To configure RPM probes from the J-Web user interface:

1. Click RPM from the Quick Configuration panel on the left side of the Configuration tab of the J-Web GUI. The Quick Configuration RPM window appears.

2. Click Add to add a probe owner. Alternatively, if probe owners are already created, click one of the probe owner names to add a new probe to the probe owner.

3. On the subsequent page, enter a name in the Owner Name field under Identification if you want to create a new probe owner. However, if you clicked one of the existing owner names in the previous step, J-Web does not allow you to change the owner name. Click Add under Performance Probe Test. The J-Web GUI displays the performance probe test parameters.

4. Configure the probe test parameters as explained in Table 5, and click OK.

To monitor probe tests configured on the device:

1. Click RPM from the left pane of the Monitor tab. The RPM Currently Running Tests page appears.

2. Click the Graph link corresponding to the test that you wish to monitor. J-Web returns a graphical representation of the test results.

**Note:** J-Web does not support RPM for EX Series switches.

For more information about using the J-Web GUI, see the **J-Web Interface User Guide**.

Figure 7 shows the metrics obtained from J-Web for the probe belonging to the Best Effort traffic forwarding class.

---

**Figure 7: Probe results from J-Web**
Figure 8 shows the RPM Event View available on J-Web.

Figure 8: RPM event viewer on J-Web
Description and Deployment Scenario

Scenario 1

Table 10 shows a sample RPM configuration snippet with an explanation of each statement. The configuration snippet consists of a probe owned by BE-ES1 that contains a test called BE-ICMP-Ping-Test. The probe type configured in this case is icmp-ping-timestamp, and it is assigned to the Best Effort forwarding class. The probes are destined for the lt interface on the local router whose loopback address is 11.1.1.192. The acceptable threshold values for RTT jitter, egress, and ingress times for the probes are set.

Table 10: RPM Configuration Snippet

| services {} | Parent object for all services. |
| rpm {} | Root object for all Real-Time Performance Monitoring Service configurations. |
| probe BE-ES1 {} | Defines a probe owner as BE-ES1. |
| test BE-ICMP-Ping-Test {} | Creates a test (collection of probes) called BE-ICMP-Ping-Test. |
| probe-type icmp-ping-timestamp; | Specifies the type of probe. |
| target address 11.1.1.191; | Sets the destination of this probe. |
| probe-interval 15; | Sets how often this probe will be sent (time interval between probes). |
| test-interval 1; | Time interval between each RPM test (series of probes). |
| source-address 11.1.1.192; | Source address of probe (if this is not specified, the probe uses the outgoing interface address as the source). |
| dscp-code-points be; | Sets DSCP bits corresponding to best effort traffic class in the IP header. |
| thresholds {} | Assigns test and probe limits. |
| rtt 60; | Sets upper limit of RTT per probe. Exceeding this value triggers a probe failure. |
| jitter-ingress 10; | Sets “10” as the threshold of the difference between the minimum and maximum outbound times. Triggers a failure if threshold is exceeded. |
| jitter-egress 10; | Sets “10” as the threshold of the difference between the maximum and minimum inbound times. Triggers a failure if threshold is exceeded. |
| traps [ rtt-exceeded test-completion jitter-exceeded ]; | Generates SNMP traps when specified thresholds are exceeded. |
| destination-interface lt-0/0/0.0; | Specifies logical tunnel (lt) interface as destination on local node. All probes will be pushed out via destination interface first regardless of actual next-hop to destination. |
| hardware-timestamp; | Enables hardware-based timestamping of RPM probe messages on router. Note: In case of J Series, the fwdd process does this since services hardware is “simulated.” |
| one-way-hardware-timestamp; | Uses all four timestamps in the packet so that it’s possible to measure one-way propagation characteristics to and from the probe target. |

Table 11 illustrates the lt interface configuration on a J Series router.
Table 11: lt Interface Configuration Snipp

interfaces {
  lt-0/0/0 {
    unit 0 {
      Logical unit 0 on lt interface
      encapsulation frame-relay;
      Define encapsulation
      dlci 10;
      Define data-link connection identifier (DLCI) on unit
      peer-unit 1;
      Remote end unit with which this unit will peer
      family inet;
      Support IPv4
    }
    unit 1 {
      Logical unit 1 on lt interface
      encapsulation frame-relay;
      Defines encapsulation
      dlci 10;
      Defines DLCI on unit
      peer-unit 0;
      Remote end unit with which this unit will peer
      family inet;
      Support IPv4
    }
  }
}

Table 12 shows a snippet of the CoS configuration where the different queues are defined, and the lt interface is defined as a part of the CoS module.

Table 12: CoS Configuration Snippet

forwarding-classes {
  queue 0 BE;
  Defines best effort queue
  queue 1 EF;
  Defines expedited forwarding queue
  queue 2 AF;
  Defines assured forwarding queue
  queue 3 NC;
  Defines network control queue
}

lt-* {
  Defines logical tunnel Interface
  unit *
    Logical unit on “lt interface”
  scheduler-map SCHED-MAP;
    CoS scheduling map to be applied
  classifiers {
    dscp DSCP-CLASSIFIER;
    DSCP classification
  }
  }
}

CLI command outputs that can be executed on the services router Session Description Protocol (SDP) are provided below. The important result fields obtained from the output of the probe have been highlighted in bold.

The peak-to-peak values are calculated for every type of measurement. The results of the CLI command are organized to display the results from the current test, then the previous test, and finally all tests. The owner of the probe is BE-ES1 and the probe type is an icmp-ping-timestamp. The test consists of a single one-way-hardware timestamped probe.
show services rpm probe-results
Owner: BE-ES1, Test: BE-ICMP-Ping-Test
Target address: 11.1.1.191, Source address: 11.1.1.192,
Probe type: icmp-ping-timestamp
Destination interface name: lt-0/0/0.0
Test size: 1 probes
Probe results:
Response received, Mon Dec  8 15:54:42 2008,
Client and server hardware timestamps
Rtt: 514 usec, Egress jitter: -1 usec, Ingress jitter: -1 usec,
Round trip jitter: -2 usec, Egress interarrival jitter: 61 usec,
Ingress interarrival jitter: 61 usec,
Round trip interarrival jitter: 88 usec
Results over current test:
Probes sent: 1, Probes received: 1, Loss percentage: 0
Measurement: Round trip time
Samples: 1, Minimum: 514 usec, Maximum: 514 usec, Average: 514 usec,
Peak to peak: 0 usec, Stddev: 0 usec, Sum: 514 usec
Measurement: Negative egress jitter
Samples: 1, Minimum: 1 usec, Maximum: 1 usec, Average: 1 usec,
Peak to peak: 0 usec, Stddev: 0 usec, Sum: 1 usec
Measurement: Negative ingress jitter
Samples: 1, Minimum: 1 usec, Maximum: 1 usec, Average: 1 usec,
Peak to peak: 0 usec, Stddev: 0 usec, Sum: 1 usec
Measurement: Negative round trip jitter
Samples: 1, Minimum: 2 usec, Maximum: 2 usec, Average: 2 usec,
Peak to peak: 0 usec, Stddev: 0 usec, Sum: 2 usec
Results over last test:
Probes sent: 1, Probes received: 1, Loss percentage: 0
Test completed on Mon Dec  8 15:54:42 2008
Measurement: Round trip time
Samples: 1, Minimum: 514 usec, Maximum: 514 usec, Average: 514 usec,
Peak to peak: 0 usec, Stddev: 0 usec, Sum: 514 usec
Measurement: Negative egress jitter
Samples: 1, Minimum: 1 usec, Maximum: 1 usec, Average: 1 usec,
Peak to peak: 0 usec, Stddev: 0 usec, Sum: 1 usec
Measurement: Negative ingress jitter
Samples: 1, Minimum: 1 usec, Maximum: 1 usec, Average: 1 usec,
Peak to peak: 0 usec, Stddev: 0 usec, Sum: 1 usec
Measurement: Negative round trip jitter
Samples: 1, Minimum: 2 usec, Maximum: 2 usec, Average: 2 usec,
Peak to peak: 0 usec, Stddev: 0 usec, Sum: 2 usec
Results over all tests:
Probes sent: 5838, Probes received: 5835, Loss percentage: 0
Measurement: Round trip time
Samples: 5835, Minimum: 433 usec, Maximum: 10472 usec,
Average: 662 usec, Peak to peak: 10039 usec, Stddev: 180 usec,
Sum: 3863754 usec
Measurement: Ingress delay
Peak to peak: 0 usec, Stddev: 0 usec
Measurement: Egress delay
Samples: 1, Minimum: 9000 usec, Maximum: 9000 usec, Average: 9000 usec,
Peak to peak: 0 usec, Stddev: 0 usec
Measurement: Positive egress jitter
Samples: 2682, Minimum: 0 usec, Maximum: 317 usec, Average: 68 usec,
Peak to peak: 317 usec, Stddev: 68 usec, Sum: 181841 usec
Measurement: Negative egress jitter
Samples: 3150, Minimum: 1 usec, Maximum: 311 usec, Average: 58 usec,
Peak to peak: 310 usec, Stddev: 58 usec, Sum: 181119 usec
Measurement: Positive ingress jitter
Samples: 3215, Minimum: 0 usec, Maximum: 301 usec, Average: 58 usec,
Peak to peak: 301 usec, Stddev: 60 usec, Sum: 185034 usec
Measurement: Negative ingress jitter
Samples: 2617, Minimum: 1 usec, Maximum: 314 usec, Average: 71 usec,
Peak to peak: 313 usec, Stddev: 68 usec, Sum: 185809 usec
Measurement: Positive round trip jitter
Samples: 3057, Minimum: 0 usec, Maximum: 324 usec, Average: 85 usec,
Peak to peak: 324 usec, Stddev: 117 usec, Sum: 259570 usec

Measurement: Negative round trip jitter
Samples: 2776, Minimum: 1 usec, Maximum: 9703 usec, Average: 97 usec,
Peak to peak: 9702 usec, Stddev: 218 usec, Sum: 269526 usec

The output from the CLI command [show services rpm history-results owner BE-ES1] displays the RPM RTT history of the probe BE-ES1 [Test = BE-ICMP-Ping-Test].

J6350-ES-2> show services rpm history-results owner BE-ES1

<table>
<thead>
<tr>
<th>Owner, Test</th>
<th>Probe received</th>
<th>Round trip time</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:42 2008</td>
<td>784 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:43 2008</td>
<td>762 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:44 2008</td>
<td>768 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:45 2008</td>
<td>536 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:46 2008</td>
<td>516 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:47 2008</td>
<td>770 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:48 2008</td>
<td>520 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:49 2008</td>
<td>520 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:50 2008</td>
<td>770 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:51 2008</td>
<td>770 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:52 2008</td>
<td>767 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:53 2008</td>
<td>764 usec</td>
</tr>
<tr>
<td>BE-ES1, BE-ICMP-Ping-Test</td>
<td>Mon Dec 8 15:55:54 2008</td>
<td>765 usec</td>
</tr>
</tbody>
</table>

One example of an RPM deployment scenario would be a case where the jitter and delay for VoIP traffic is computed using probes that contain a stream of ICMP/TCP/UDP packets. Similarly, HTTP get is a protocol that can be used for application performance measurement. The user can select the type of packets that are contained in the probe based on the metrics that need to be collected.

Scenario 2
ICMP PING PROBE on M Series and T Series

Topology

Step 1: Here’s the server configuration.

```
sp-1/3/0 {
  unit 0 {
    family inet;
  }
  unit 1 {
    rpm server;
    family inet {
      address 3.3.3.3/32;
    }
  }
}
```
Step 2: Here's the interface configuration on the client.

```plaintext
sp-1/2/0 {
    services-options {
        syslog {
            host local {
                services any;
            }
        }
    }
    unit 0 {
        family inet;
    }
    unit 10 {
        rpm client;
        family inet {
            address 1.1.1.1/32;
        }
    }
}
```

Step 3: Here are the SNMP commands to create the RPM entry.

```
snmpset -Os eddy fred pingCtlRowStatus.1.100.1.100 i createAndWait
snmpset -Os eddy fred pingCtlTargetAddressType.1.100.1.100 i ipv4
snmpset -Os eddy fred pingCtlTargetAddress.1.100.1.100 x 03030303
snmpset -Os eddy fred pingCtlProbeCount.1.100.1.100 u 10
snmpset -Os eddy fred pingCtlFrequency.1.100.1.100 u 60
snmpset -Os eddy fred pingCtlTimeOut.1.100.1.100 u 1
snmpset -Os eddy fred jnxPingCtlIfName.1.100.1.100 s "sp-1/2/0.10"
snmpset -Os eddy fred jnxPingCtlOneWayHWTimestamp.1.100.1.100 i true
snmpset -Os eddy fred jnxPingCtlMovAvgSize.1.100.1.100 u 25
snmpset -Os eddy fred pingCtlByPassRouteTable.1.100.1.100 i true
snmpset -Os eddy fred pingCtlRowStatus.1.100.1.100 i active
snmpset -Os eddy fred pingCtlAdminStatus.1.100.1.100 i enabled
```

Step 4: Here's the CLI output.

```
root@eddy> show services rpm probe-results owner d
Owner: d, Test: d
Target address: 3.3.3.3, Probe type: icmp-ping
Destination interface name: sp-1/2/0.10
Test size: 10 probes
Probes results:
    Response received, Fri Mar  9 13:42:38 2007,
    Client and server hardware timestamps
    Rtt: 118 usec, Egress jitter: 5 usec, Ingress jitter: -2 usec,
    Round trip jitter: 2 usec, Egress interarrival jitter: 5 usec,
    Ingress interarrival jitter: 4 usec,
    Round trip interarrival jitter: 3 usec
Results over current test:
    Probes sent: 10, Probes received: 10, Loss percentage: 0
    Measurement: Round trip time
        Samples: 10, Minimum: 11 usec, Maximum: 118 usec, Average: 114 usec,
        Peak to peak: 7 usec, Stddev: 2 usec, Sum: 1138 usec
    Measurement: Positive egress jitter
        Samples: 6, Minimum: 1 usec, Maximum: 11 usec, Average: 4 usec,
        Peak to peak: 10 usec, Stddev: 3 usec, Sum: 23 usec
    Measurement: Negative egress jitter
        Samples: 3, Minimum: 2 usec, Maximum: 10 usec, Average: 6 usec,
        Peak to peak: 8 usec, Stddev: 3 usec, Sum: 18 usec
```
Measurement: Positive ingress jitter
Samples: 4, Minimum: 1 usec, Maximum: 6 usec, Average: 3 usec,
Peak to peak: 5 usec, Stdev: 2 usec, Sum: 13 usec
Measurement: Negative ingress jitter
Samples: 5, Minimum: 1 usec, Maximum: 9 usec, Average: 3 usec,
Peak to peak: 8 usec, Stdev: 3 usec, Sum: 16 usec
Measurement: Positive round trip jitter
Samples: 6, Minimum: 1 usec, Maximum: 3 usec, Average: 2 usec,
Peak to peak: 2 usec, Stdev: 1 usec, Sum: 12 usec
Measurement: Negative round trip jitter
Samples: 4, Minimum: 1 usec, Maximum: 4 usec, Average: 2 usec,
Peak to peak: 3 usec, Stdev: 1 usec, Sum: 7 usec

Results over last test:
Probes sent: 10, Probes received: 10, Loss percentage: 0
Test completed on Fri Mar  9 13:42:38 2007
Measurement: Round trip time
Samples: 10, Minimum: 111 usec, Maximum: 118 usec, Average: 114 usec,
Peak to peak: 7 usec, Stdev: 2 usec, Sum: 1138 usec
Measurement: Positive egress jitter
Samples: 3, Minimum: 2 usec, Maximum: 11 usec, Average: 4 usec,
Peak to peak: 10 usec, Stdev: 3 usec, Sum: 23 usec
Measurement: Negative egress jitter
Samples: 4, Minimum: 1 usec, Maximum: 6 usec, Average: 3 usec,
Peak to peak: 5 usec, Stdev: 2 usec, Sum: 13 usec
Measurement: Positive ingress jitter
Samples: 5, Minimum: 1 usec, Maximum: 9 usec, Average: 3 usec,
Peak to peak: 8 usec, Stdev: 3 usec, Sum: 16 usec
Measurement: Positive round trip jitter
Samples: 6, Minimum: 1 usec, Maximum: 3 usec, Average: 2 usec,
Peak to peak: 2 usec, Stdev: 1 usec, Sum: 12 usec
Measurement: Negative round trip jitter
Samples: 4, Minimum: 1 usec, Maximum: 4 usec, Average: 2 usec,
Peak to peak: 3 usec, Stdev: 1 usec, Sum: 7 usec

Results over moving average:
Probes sent: 25, Probes received: 25, Loss percentage: 0
Measurement: Round trip time
Samples: 25, Minimum: 110 usec, Maximum: 120 usec, Average: 114 usec,
Peak to peak: 10 usec, Stdev: 3 usec, Sum: 2861 usec
Measurement: Positive egress jitter
Samples: 12, Minimum: 1 usec, Maximum: 12 usec, Average: 5 usec,
Peak to peak: 11 usec, Stdev: 4 usec, Sum: 65 usec
Measurement: Negative egress jitter
Samples: 11, Minimum: 1 usec, Maximum: 10 usec, Average: 4 usec,
Peak to peak: 9 usec, Stdev: 3 usec, Sum: 45 usec
Measurement: Positive ingress jitter
Samples: 10, Minimum: 1 usec, Maximum: 6 usec, Average: 3 usec,
Peak to peak: 5 usec, Stdev: 2 usec, Sum: 28 usec
Measurement: Negative ingress jitter
Samples: 13, Minimum: 1 usec, Maximum: 12 usec, Average: 4 usec,
Peak to peak: 11 usec, Stdev: 3 usec, Sum: 54 usec
Measurement: Positive round trip jitter
Samples: 14, Minimum: 0 usec, Maximum: 10 usec, Average: 3 usec,
Peak to peak: 10 usec, Stdev: 3 usec, Sum: 46 usec
Measurement: Negative round trip jitter
Samples: 11, Minimum: 1 usec, Maximum: 8 usec, Average: 4 usec,
Peak to peak: 7 usec, Stdev: 3 usec, Sum: 41 usec

Results over all tests:
Probes sent: 1870, Probes received: 1870, Loss percentage: 0
Measurement: Round trip time
Samples: 1870, Minimum: 102 usec, Maximum: 134 usec, Average: 114 usec,
Peak to peak: 32 usec, Stdev: 4 usec, Sum: 213438 usec
Measurement: Positive egress jitter
Samples: 1081, Minimum: 0 usec, Maximum: 319 usec, Average: 7 usec,
Peak to peak: 319 usec, Stdev: 30 usec, Sum: 7053 usec
Step 5: Here’s the SNMP output.

```
kyum-bsd# snmpwalk -Os eddy public jnxRpmResultsSampleTable
jnxRpmResSampleValue."d"."d".roundTripTime = 118
jnxRpmResSampleValue."d"."d".rttJitter = 2
jnxRpmResSampleValue."d"."d".rttInterarrivalJitter = 3
jnxRpmResSampleValue."d"."d".egressJitter = 5
jnxRpmResSampleValue."d"."d".egressInterarrivalJitter = 5
jnxRpmResSampleValue."d"."d".ingressJitter = -2
jnxRpmResSampleValue."d"."d".ingressInterarrivalJitter = 4
jnxRpmResSampleTsType."d"."d".roundTripTime = clientAndServerHardware(3)
jnxRpmResSampleTsType."d"."d".rttJitter = clientAndServerHardware(3)
jnxRpmResSampleTsType."d"."d".rttInterarrivalJitter = clientAndServerHardware(3)
jnxRpmResSampleTsType."d"."d".egressJitter = clientAndServerHardware(3)
jnxRpmResSampleTsType."d"."d".egressInterarrivalJitter = clientAndServerHardware(3)
jnxRpmResSampleTsType."d"."d".ingressJitter = clientAndServerHardware(3)
jnxRpmResSampleTsType."d"."d".ingressInterarrivalJitter = clientAndServerHardware(3)
jnxRpmResSampleDate."d"."d".roundTripTime = 2007-3-9,13:42:38.0,-8:0
jnxRpmResSampleDate."d"."d".rttJitter = 2007-3-9,13:42:38.0,-8:0
jnxRpmResSampleDate."d"."d".rttInterarrivalJitter = 2007-3-9,13:42:38.0,-8:0
jnxRpmResSampleDate."d"."d".egressJitter = 2007-3-9,13:42:38.0,-8:0
jnxRpmResSampleDate."d"."d".egressInterarrivalJitter = 2007-3-9,13:42:38.0,-8:0
jnxRpmResSampleDate."d"."d".ingressJitter = 2007-3-9,13:42:38.0,-8:0
jnxRpmResSampleDate."d"."d".ingressInterarrivalJitter
```

Supported Routing and Switching Platforms

**Hardware**

Real-time performance monitoring is supported on the following Juniper Networks routing and switching platforms:

- Juniper Networks EX3200 Ethernet Switch
- Juniper Networks EX4200 Ethernet Switch
- J Series Services Routers
- M Series Multiservice Edge Routers
- Juniper Networks MX Series Ethernet Services Routers
- T Series Core Routers
- Juniper Networks SRX Series Services Gateways
Summary

This document has provided an overview of RPM features in Juniper Networks JUNOS Software. It has also provided information about tools that you can use for real-time performance monitoring on devices running JUNOS. This document provides guidance for integration of RPM into any Third party management systems for SLA and performance monitoring.

The main benefits of enabling RPM on JUNOS devices include:

- Measurement of network performance
- SLA monitoring
- Capacity planning
- Fault isolation and network analysis before and after the deployment of services
- KPIs such as network latency and jitter

For more information about real-time performance monitoring tools and features supported by JUNOS, see the following documents at http://www.juniper.net/techpubs/software/junos/:

- JUNOS Network Management Configuration Guide
- JUNOS Services Interfaces Configuration Guide
- J-Web Interface User Guide

About Juniper Networks

Juniper Networks, Inc. is the leader in high-performance networking. Juniper offers a high-performance network infrastructure that creates a responsive and trusted environment for accelerating the deployment of services and applications over a single network. This fuels high-performance businesses. Additional information can be found at www.juniper.net.

Corporate and Sales Headquarters
Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or 408.745.2000
Fax: 408.745.2100

APAC Headquarters
Juniper Networks (Hong Kong)
26/F, Cityplaza One
1111 King’s Road
Taikoo Shing, Hong Kong
Phone: 852.2332.3636
Fax: 852.2574.7803

EMEA Headquarters
Juniper Networks Ireland
Airside Business Park
Swords, County Dublin,
Ireland
Phone: 35.31.8903.600
Fax: 35.31.8903.601

Copyright 2009 Juniper Networks, Inc.
All rights reserved. Juniper Networks, the
Juniper Networks logo, JUNOS, NetScreen,
and ScreenOS are registered trademarks of
Juniper Networks, Inc. in the United States
and other countries. JUNOSe is a trademark of
Juniper Networks, Inc. All other trademarks,
service marks, registered marks, or registered
service marks are the property of their
respective owners. Juniper Networks assumes
no responsibility for any inaccuracies in this
document. Juniper Networks reserves the right
to change, modify, transfer, or otherwise revise
this publication without notice.

To purchase Juniper Networks solutions, please contact your Juniper Networks representative at 1-866-298-6428 or authorized reseller.