Engineering Simplicity

Junos® OS

Monitoring, Sampling, and Collection Services Interfaces User Guide

Published
2025-12-16

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

Junos® OS Monitoring, Sampling, and Collection Services Interfaces User Guide

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

Table of Contents

About This Guide

Use this guide to configure traffic flow monitoring, packet flow capture, inline monitoring, traffic sampling for accounting or discard, real-time performance monitoring (RPM and TWAMP), RFC 2544 performance benchmarking, and inline video monitoring.

1

PART

Flow Monitoring and Flow Collection Services

CHAPTER 1

Understanding Flow Monitoring

IN THIS CHAPTER

Flow Monitoring Terms and Acronyms

IN THIS SECTION

active flow monitoring

Use this technique to lawfully intercept and observe specified data network traffic on an active router participating in the network.

Adaptive Services PIC

This advanced PIC delivers active flow monitoring, Network Address Translation (NAT), stateful firewall, and intrusion detection functions. For more information about the Adaptive Services PIC, see the *Junos Services Interfaces Configuration Guide*.

cflowd

The version 5 and version 8 flow monitoring process that captures flow information from network traffic and exports this data into summary tables. Once the system is finished capturing the flow data, you can analyze the data as needed. For more information about `cflowd`, see **http://www.caida.org**.

content destination

A recipient that receives monitored packets from a DTCP or dynamic flow capture-enabled monitoring station.

control source

A control source is a dynamic flow capture client that monitors electronic data or voice transfer over the network. The control source uses DTCP to send filter requests to the dynamic flow capture-enabled monitoring station.

dynamic flow capture

Use this technique to enable DTCP-enabled control sources to send specified filtering criteria in real time to a monitoring station. The monitoring station passively monitors the specified traffic flows on demand and sends the captured packets to content destinations.

DTCP (Dynamic Tasking Control Protocol)

Use this protocol to specify filtering criteria in a dynamic flow capture environment.

ES PIC

This PIC provides encryption and security services (such as IPsec).

flow collector interface

A converted Monitoring Services II PIC that processes multiple flow records into compressed ASCII data files and exports these files to an FTP server.

Monitoring Services PIC

The original PIC that provides passive and active flow monitoring.

Monitoring Services II PIC

An advanced PIC that provides passive flow monitoring.

Monitoring Services III PIC

An advanced PIC that provides dynamic flow capture.

Multiservices 100 PIC

Also referred to as MS PIC Type 1, an advanced PIC that provides active flow capture.

Multiservices 400 PIC

Also referred to as MS PIC Type 2, an advanced PIC that provides active flow capture.

Multiservices 500 PIC

Also referred to as MS PIC Type 3, an advanced PIC that handles active flow capture.

passive flow monitoring

Use this technique to lawfully intercept and observe specified data network traffic on a passive flow monitoring station that is not participating in the network.

Configuring Flow Monitoring

IN THIS SECTION

The flow-monitoring application performs traffic flow monitoring and enables lawful interception of traffic between two routers or switches. Traffic flows can either be passively monitored by an offline router or switch or actively monitored by a router participating in the network.

Configuring Flow-Monitoring Interfaces

To enable flow monitoring on the Monitoring Services PIC, include the `mo-`*`fpc/pic/port`* statement at the `[edit interfaces]` hierarchy level:

```
mo-fpc/pic/port {
    unit logical-unit-number {
        family inet {
            address address {
                destination address;
            }
            filter {
                group filter-group-number;
                input filter-name;
                output filter-name;
            }
            sampling {
                [ input output ];
            }
        }
    }
    multiservice-options {
        (core-dump | no-core-dump);
        (syslog | no-syslog);
        flow-control-options {
```

```
            down-on-flow-control;
            dump-on-flow-control;
            reset-on-flow-control;
        }
    }
}
```

Specify the physical and logical location of the flow-monitoring interface. You cannot use `unit 0`, because it is already used by internal processes. Specify the source and destination addresses. The `filter` statement allows you to associate an input or output filter or a filter group that you have already configured for this purpose. The `sampling` statement specifies the traffic direction: `input`, `output`, or both.

The `multiservice-options` statement allows you to configure properties related to flow-monitoring interfaces:

- Include the `core-dump` statement to enable storage of core files in **/var/tmp**.
- Include the `syslog` statement to enable storage of system logging information in **/var/log**.

> **NOTE**: Boot images for monitoring services interfaces are specified at the `[edit chassis images pic]` hierarchy level. You must include the following configuration to make the flow monitoring feature operable:
>
> ```
> [edit system]
> ntp {
>     boot-server ntp.example.net;
>     server 172.17.28.5;
> }
> processes {
>     ntp enable;
> }
> ```

- Include the `flow-control-options` statement to configure flow control.

> **NOTE**: The multiservices PIC management daemon core file is generated when a prolonged flow control failure occurs and when you configure the setting to generate a core dump during prolonged flow control (by using the `dump-on-flow-control` option with the `flow-control-options` statement). The watchdog functionality continues to generate a kernel core file in such scenarios.

Configuring Flow-Monitoring Properties

To configure flow-monitoring properties, include the `monitoring` statement at the `[edit forwarding-options]` hierarchy level:

```
monitoring name {
    family inet {
        output {
            cflowd hostname port port-number;
            export-format format;
            flow-active-timeout seconds;
            flow-export-destination {
                collector-pic;
            }
            flow-inactive-timeout seconds;
            interface interface-name {
                engine-id number;
                engine-type number;
                input-interface-index number;
                output-interface-index number;
                source-address address;
            }
        }
    }
```

A monitoring instance is a named entity that specifies collector information under the `monitoring` *name* statement. The following sections describe the properties you can configure:

Directing Traffic to Flow-Monitoring Interfaces

To direct traffic to a flow-monitoring interface, include the `interface` statement at the `[edit forwarding-options monitoring` *name* `output]` hierarchy level. By default, the Junos OS automatically assigns values for the `engine-id` and `engine-type` statements:

- `engine-id`—Monitoring interface location.
- `engine-type`—Platform-specific monitoring interface type.

The `source-address` statement specifies the traffic source for transmission of cflowd information; you must configure it manually. If you provide a different `source-address` statement for each monitoring services output interface, you can track which interface processes a particular cflowd record.

By default, the `input-interface-index` value is the SNMP index of the input interface. You can override the default by including a specific value. The `input-interface-index` and `output-interface-index` values are exported in fields present in the cflowd version 5 flow format.

Exporting Flows

To direct traffic to a flow collection interface, include the `flow-export-destination` statement. For more information about flow collection, see "Active Flow Monitoring Overview" on page 54.

To configure the cflowd version number, include the `export-format` statement at the `[edit forwarding-options monitoring` *`name`* `output]` hierarchy level. By default, version 5 is used. Version 8 enables the router software to aggregate the flow information using broader criteria and reduce cflowd traffic. Version 8 aggregation is performed periodically (every few seconds) on active flows and when flows are allowed to expire. Because the aggregation is performed periodically, active timeout events are ignored.

For more information on cflowd properties, see "Enabling Flow Aggregation" on page 595.

Configuring Time Periods When Flow Monitoring Is Active and Inactive

To configure time periods for active flow monitoring and intervals of inactivity, include the `flow-active-timeout` and `flow-inactive-timeout` statements at the `[edit forwarding-options monitoring` *`name`* `output]` hierarchy level:

- The `flow-active-timeout` statement specifies the time interval between flow exports for active flows. If the interval between the time the last packet was received and the time the flow was last exported exceeds the configured value, the flow is exported.

  This timer is needed to provide periodic updates when a flow has a long duration. The active timeout setting enables the router to retain the start time for the flow as a constant and send out periodic cflowd reports. This in turn allows the collector to register the start time and determine that a flow has survived for a duration longer than the configured active timeout.

  > **NOTE**: In active flow monitoring, the cflowd records are exported after a time period that is a multiple of 60 seconds and greater than or equal to the configured active timeout value. For example, if the active timeout value is 90 seconds, the cflowd records are exported at 120-second intervals. If the active timeout value is 150 seconds, the cflowd records are exported at 180-second intervals, and so forth.

- The `flow-inactive-timeout` statement specifies the interval of inactivity for a flow that triggers the flow export. If the interval between the current time and the time that the last packet for this flow was received exceeds the configured inactive timeout value, the flow is allowed to expire.

If the flow stops transmitting for longer than the configured inactive timeout value, the router or switch purges it from the flow table and exports the cflowd record. As a result, the flow is forgotten as far as the PIC is concerned and if the same 5-tuple appears again, it is assigned a new start time and considered a new flow.

Both timers are necessary. The active timeout setting is needed to provide information for flows that constantly transmit packets for a long duration. The inactive timeout setting enables the router or switch to purge flows that have become inactive and that can waste tracking resources.

NOTE: The router must contain an Adaptive Services, Multiservices, or Monitoring Services PIC for the `flow-active-timeout` and `flow-inactive-timeout` statements to take effect.

Example: Configuring Flow Monitoring

The following is an example of flow-monitoring properties configured to support input interfaces, output monitoring services interfaces, and export to cflowd for flow analysis. To complete the configuration, you also need to configure the interfaces and set up a virtual private network (VPN) routing and forwarding (VRF) instance. For information on cflowd, see "Enabling Flow Aggregation" on page 595.

```
[edit forwarding-options]
monitoring group1 {
    family inet {
        output {
            cflowd 192.168.245.2 port 2055;
            export-format cflowd-version-5;
            flow-active-timeout 60;
            flow-inactive-timeout 30;
            interface mo-4/0/0.1 {
                engine-id 1;
                engine-type 1;
                input-interface-index 44;
                output-interface-index 54;
                source-address 192.168.245.1;
            }
            interface mo-4/1/0.1 {
                engine-id 2;
                engine-type 1;
                input-interface-index 45;
                output-interface-index 55;
                source-address 192.168.245.1;
            }
```

```
            interface mo-4/2/0.1 {
                engine-id 3;
                engine-type 1;
                input-interface-index 46;
                output-interface-index 56;
                source-address 192.168.245.1;
            }
            interface mo-4/3/0.1 {
                engine-id 4;
                engine-type 1;
                input-interface-index 47;
                output-interface-index 57;
                source-address 192.168.245.1;
            }
        }
    }
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
15.1	The multiservices PIC management daemon core file is generated when a prolonged flow control failure occurs and when you configure the setting to generate a core dump during prolonged flow control (by using the `dump-on-flow-control` option with the `flow-control-options` statement).

RELATED DOCUMENTATION

Active Flow Monitoring Overview | 54

Directing Replicated Flows from M and T Series Routers to Multiple Flow Servers | 658

Configuring Services Interface Redundancy with Flow Monitoring | 72

Example: Configure Active Monitoring on a Logical System for MX Series and PTX Series Routers | 58

Flow Monitoring Output Formats

When you implement passive flow monitoring and active flow monitoring, you should be familiar with flow monitoring formats and fields. Version 5 and version 8 export data into specified fields. Version 9 exports data into templates.

The flow monitoring station monitors the traffic flow and exports the data in flow format to an external server. The Junos OS collects information about the following fields:

- Source and destination IP address
- Total number of bytes and packets sent
- Start and end times of the data flow
- Source and destination port numbers
- TCP flags
- IP protocol and IP type of service
- Originating AS of source and destination address
- Source and destination address prefix mask lengths
- Next-hop router’s IP address
- MPLS label (version 9 only)
- ICMP (version 9 only)

Detailed descriptions of the formats are available as follows:

- "Flow Monitoring Version 5 Format Output Fields" on page 11
- "Flow Monitoring Version 8 Format Output Fields" on page 16
- "Flow Monitoring Version 9 Format Output Fields" on page 26

Flow Monitoring Version 5 Format Output Fields

A detailed explanation of version 5 packet formats and fields is shown in the following figures and tables:

- Figure 1 on page 12

- Table 1 on page 12
- Figure 2 on page 13
- Table 2 on page 14

Figure 1: Version 5 Packet Header Format

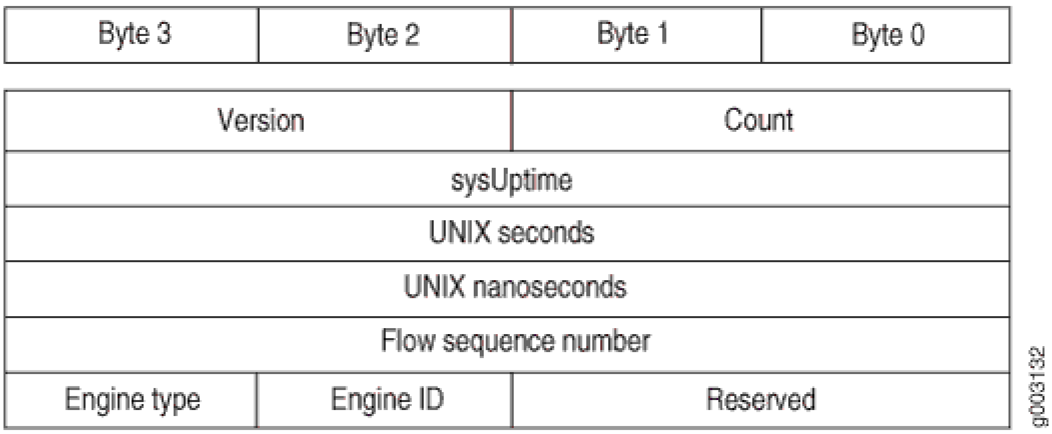


Table 1: Export Version 5 Packet Header Fields

Field	Description	Comments
`Version`	5	–
`Count`	The number of records in the Protocol Data Unit (PDU) or packet	–
`sysUptime`	Current time elapsed, in milliseconds, since the router started	–
`UNIX seconds`	Current seconds since 0000 UTC 1970	NTP synchronized time; the clock on each services PIC is autonomous (200–400 msec jitter) across PICs in a chassis
`UNIX nanoseconds`	Residual nanoseconds since 0000 UTC 1970	See Comments above for UNIX seconds

Table 1: Export Version 5 Packet Header Fields *(Continued)*

Field	Description	Comments
`Flow sequence number`	Sequence number of total flows received	–
`Engine type`	User-configured 8-bit value	Also known as VIP type on other vendors' equipment
`Engine ID`	User-configured 8-bit value	–

Figure 2: Version 5 Flow-Export Flow Header Format

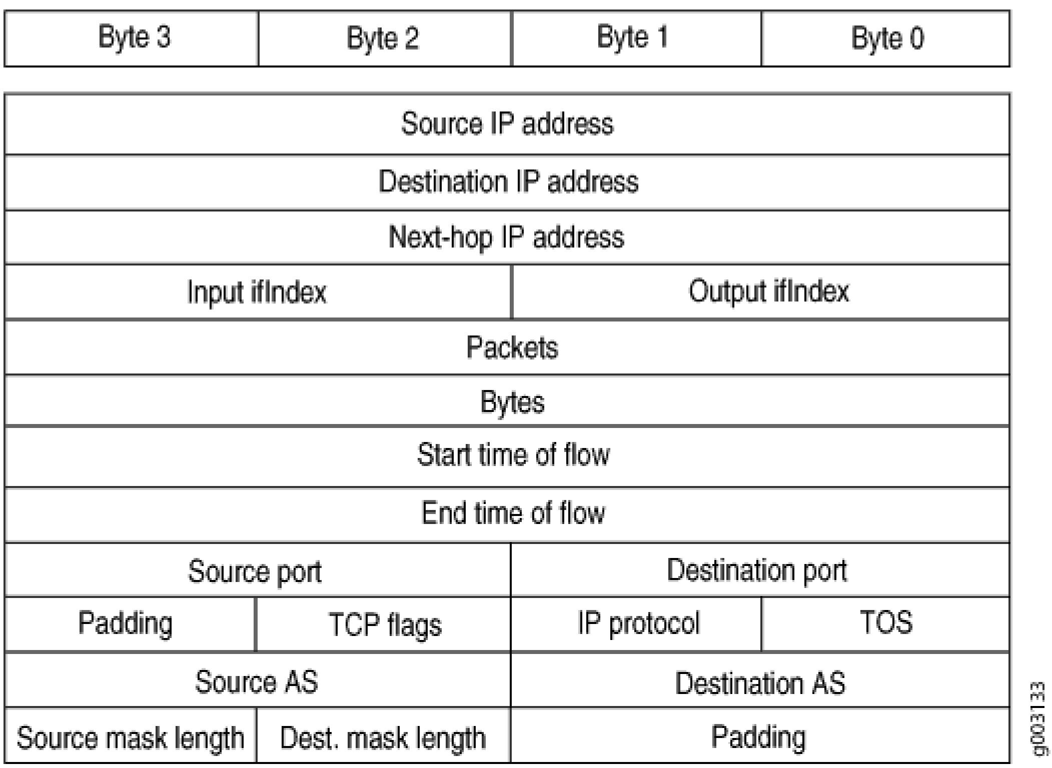


Table 2: Export Version 5 Flow-Export Flow Header Fields

Field	Description	Comments
`Source IP address`	Source IP address of the flow	–
`Destination IP address`	Destination IP address of the flow	–
`Next-hop IP address`	IP address of the router where flows are forwarded	–
`Input ifIndex`	SNMP index value for the input interface where the router receives flows	Dynamically inserted, but overridden by manual configuration
`Output ifIndex`	SNMP index value for the output interface where the router forwards flows	Dynamically inserted, but overridden by manual configuration
`Packets`	Total number of packets received in a flow	–
`Bytes`	Total number of bytes received in a flow	–
`Start time of flow`	System up time, in seconds, at the start of the flow	System up time for the services PIC accepting flows
`End time of flow`	System up time, in seconds, at the end of the flow	System up time for the services PIC accepting flows
`Source port`	Source application port	–
`Destination port`	Destination application port	The ICMP type is placed in the high-order byte and the ICMP type code is placed in the low-order byte of this field

Table 2: Export Version 5 Flow-Export Flow Header Fields *(Continued)*

Field	Description	Comments
`TCP flags`	TCP flags set in the flow	–
`IP protocol`	IP protocol number	–
`TOS`	IP type of service	–
`Source AS`	AS number of the source address	Dynamically inserted if AS information is available
`Destination AS`	AS number of the destination address	Dynamically inserted if AS information is available
`Source mask length`	Source address network mask length	–
`Dest. mask length`	Destination address network mask length	–
`Padding`	Bytes available to ensure a minimum packet length	–

Useful formulas for flow monitoring are:

- start flow timestamp absolute = *unixTime* x 1000 – (*sysUptime* – start flow timestamp)
- end flow timestamp absolute = *unixTime* x 1000 – (*sysUptime* – end flow timestamp)

NOTE: In the 2-byte destination port field of the export version 5 flow-export flow format, the following information can be derived:

- High-order byte—ICMP type
- Low-order byte—ICMP type code

For example, if the ICMP type is 3 (00000011 in binary) and the ICMP type code is network unreachable (Type Code 0, or 00000000 in binary), the resulting destination port field value is 00000011 00000000 (768 in decimal).

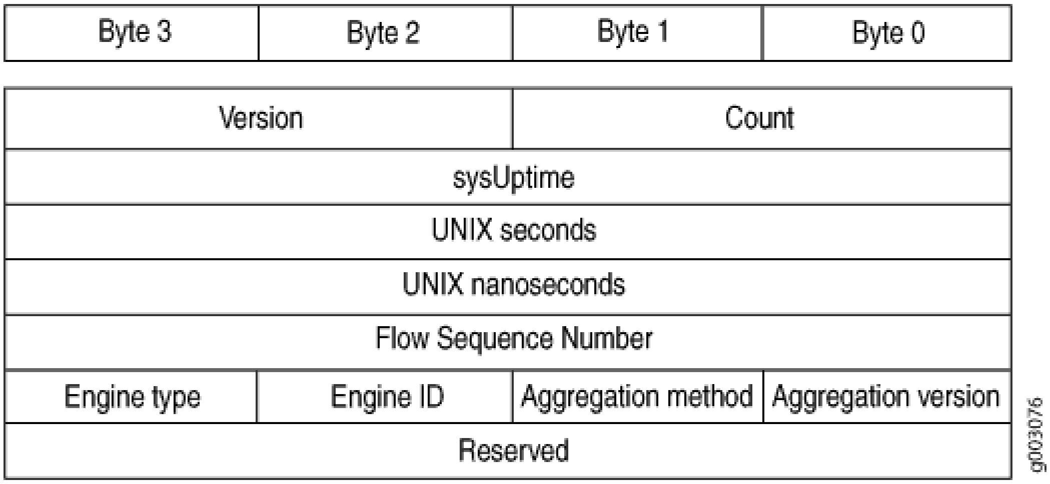
For more information on ICMP type and type code, see RFC 792 at https://www.ietf.org.

Flow Monitoring Version 8 Format Output Fields

A detailed explanation of version 8 packet formats and fields is shown as follows:

- Figure 3 on page 17
- Table 3 on page 17
- Figure 4 on page 18
- Table 4 on page 19
- Figure 5 on page 20
- Table 5 on page 20
- Figure 6 on page 22
- Table 6 on page 22
- Figure 7 on page 24
- Table 7 on page 24
- Figure 8 on page 25
- Table 8 on page 25

Figure 3: Version 8 Template Flow Format



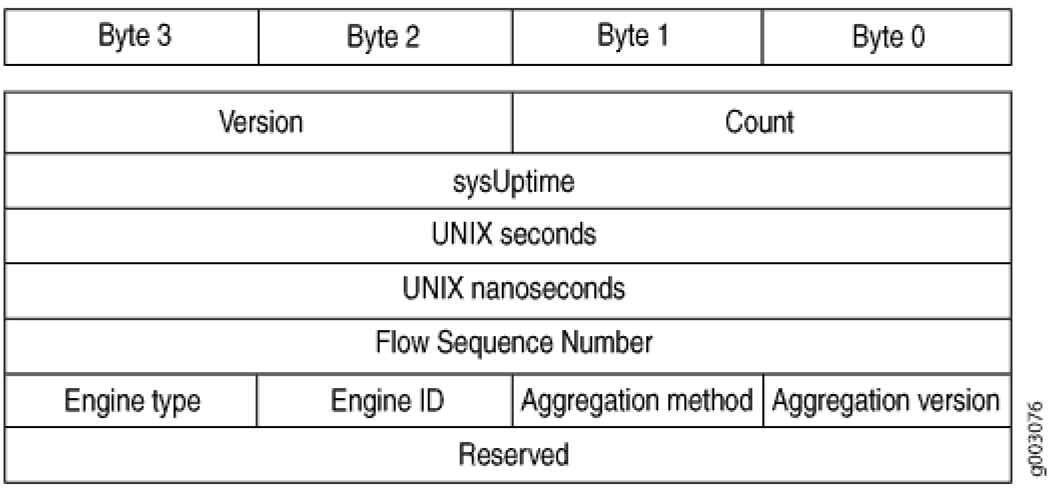


Table 3: Version 8 Flow Template Fields

Field	Description
Version	8
Count	The number of records in the protocol data unit (PDU) or packet
sysUptime	Current time elapsed, in milliseconds, since the router started

Table 3: Version 8 Flow Template Fields *(Continued)*

Field	Description
UNIX seconds	Current seconds since 0000 UTC 1970
UNIX nanoseconds	Residual nanoseconds since 0000 UTC 1970
Flow sequence number	Sequence counter of total flows received
Engine type	Type of flow switching engine
Engine ID	ID number of the flow switching engine
Aggregation method	Aggregation method used
Aggregation version	Version of the aggregation export
Reserved	Empty field reserved for future usage

Figure 4: Version 8 AS Aggregation Flow Entry Format

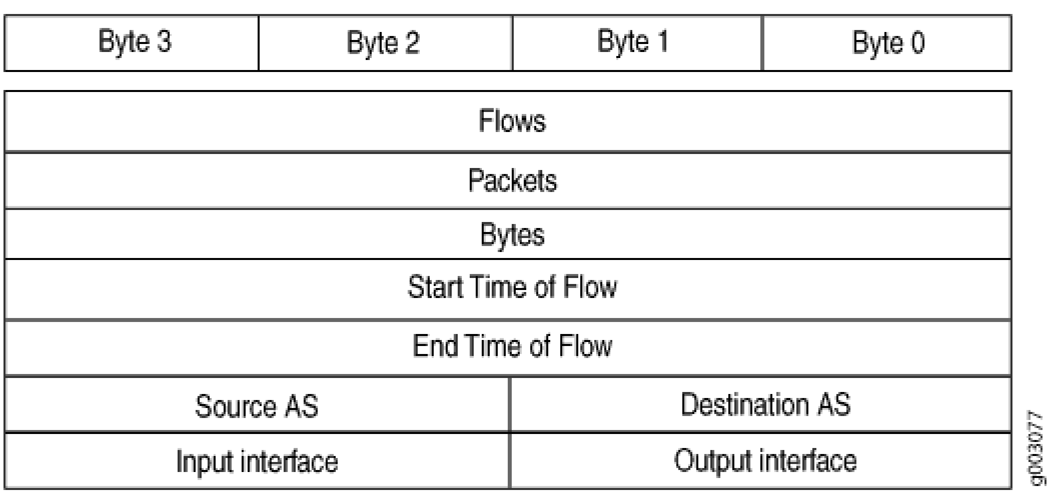


Table 4: Version 8 AS Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time, in seconds, at the start of the flow
End time of flow	System up time, in seconds, at the end of the flow
Source AS	AS number of the source address
Destination AS	AS number of the destination address
Input interface	SNMP index value for the input interface where the router receives flows
Output interface	SNMP index value for the output interface where the router forwards flows

Figure 5: Version 8 Protocol/Port Aggregation Flow Entry Format

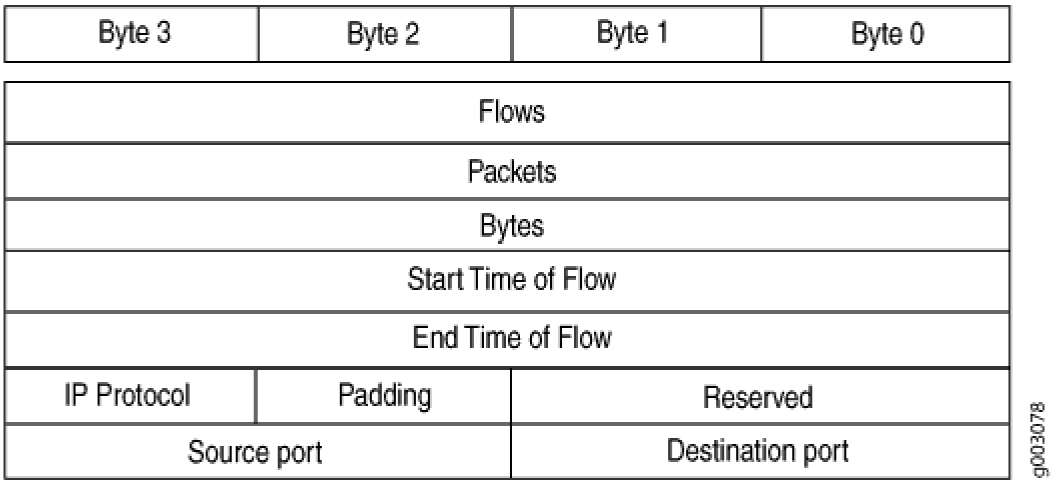


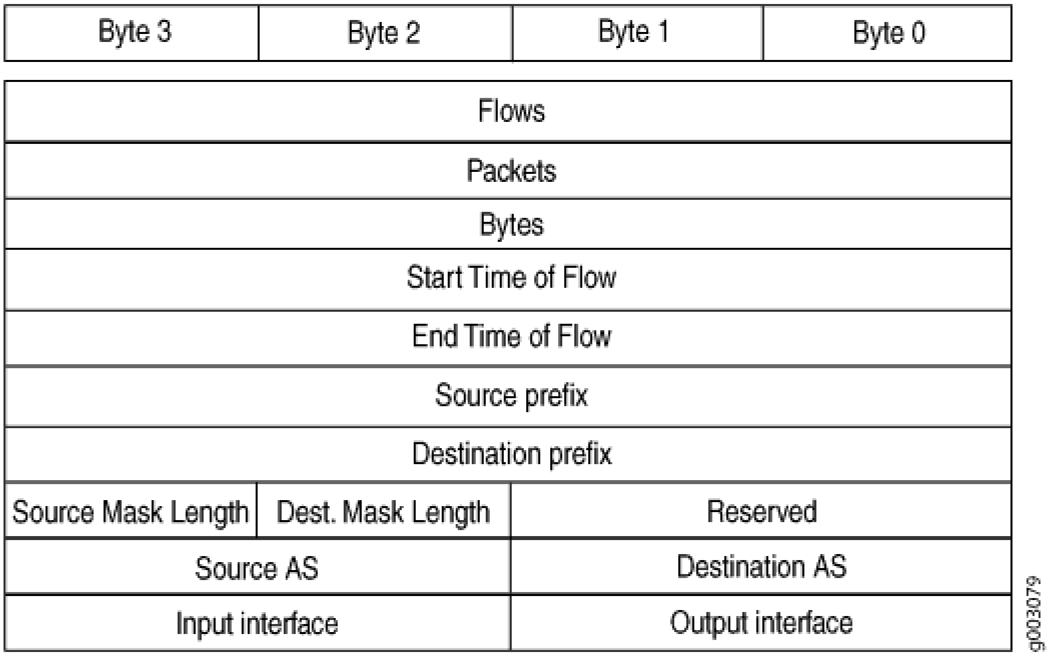
Table 5: Version 8 Protocol/Port Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time, in seconds, at the start of the flow
End time of flow	System up time, in seconds, at the end of the flow
IP protocol	IP protocol number
Padding	Bytes available to ensure a minimum packet length
Reserved	Empty field reserved for future usage
Source port	Source application port

Table 5: Version 8 Protocol/Port Aggregation Flow Entry Fields *(Continued)*

Field	Description
Destination port	Destination application port

Figure 6: Version 8 Prefix Aggregation Flow Entry Format



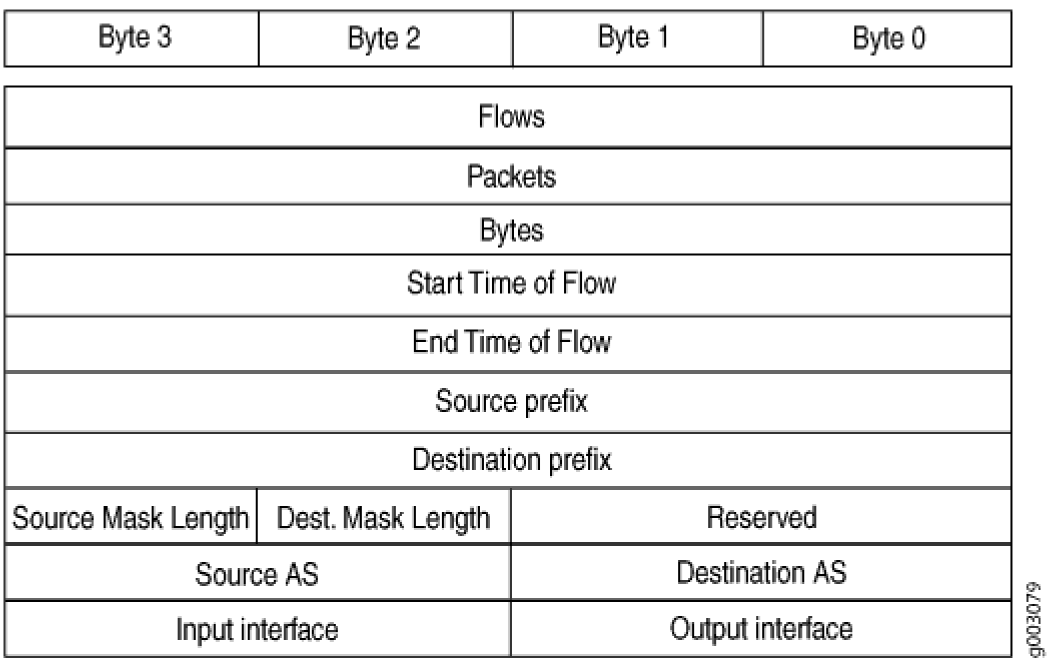


Table 6: Version 8 Prefix Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows

Table 6: Version 8 Prefix Aggregation Flow Entry Fields *(Continued)*

Field	Description
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time, in seconds, at the start of the flow
End time of flow	System up time, in seconds, at the end of the flow
Source prefix	Source IP address prefix
Destination prefix	Destination IP address prefix
Source mask length	Source address network mask length
Dest. mask length	Destination address network mask length
Reserved	Empty field reserved for future usage
Source AS	AS number of the source address
Destination AS	AS number of the destination address
Input interface	SNMP index value for the input interface where the router receives flows
Output interface	SNMP index value for the output interface where the router forwards flows

Figure 7: Version 8 Source Prefix Aggregation Flow Entry Format

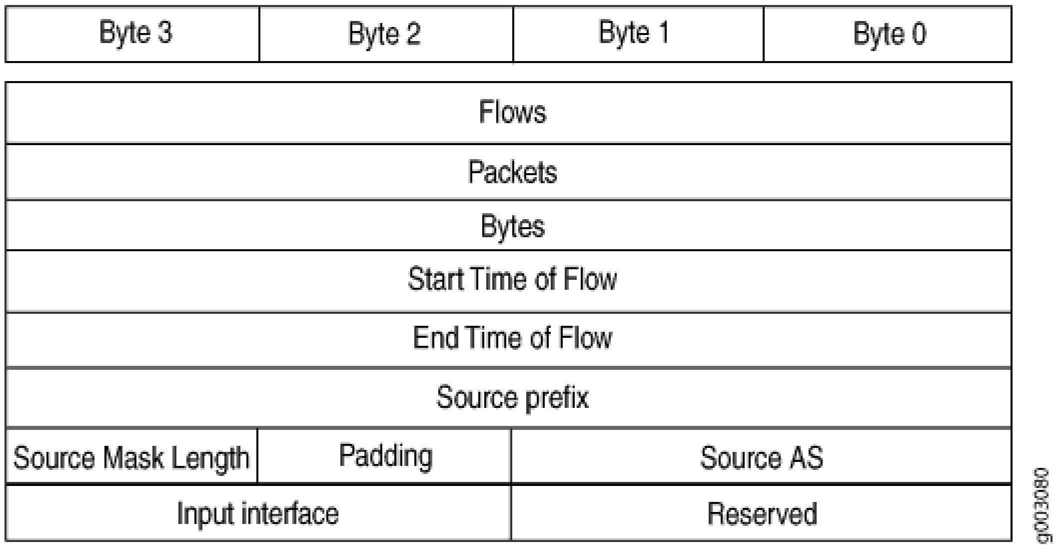


Table 7: Version 8 Source Prefix Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time, in seconds, at the start of the flow
End time of flow	System up time, in seconds, at the end of the flow
Source prefix	Source IP address prefix
Source mask length	Source address network mask length
Padding	Bytes available to ensure a minimum packet length
Source AS	AS number of the source address

Table 7: Version 8 Source Prefix Aggregation Flow Entry Fields *(Continued)*

Field	Description
Input interface	SNMP index value for the input interface where the router receives flows
Reserved	Empty field reserved for future usage

Figure 8: Version 8 Destination Prefix Aggregation Flow Entry Format

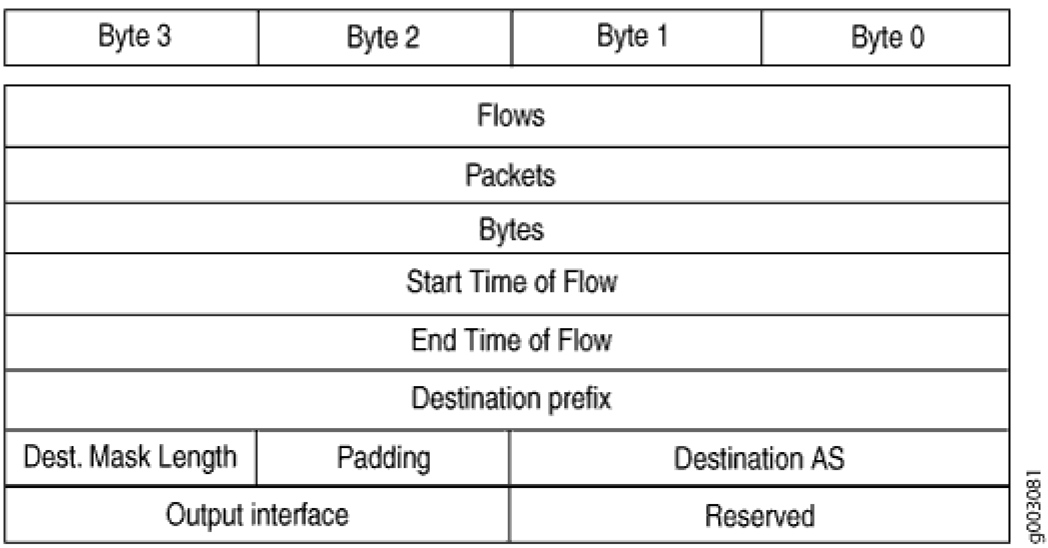


Table 8: Version 8 Destination Prefix Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time, in seconds, at the start of the flow

Table 8: Version 8 Destination Prefix Aggregation Flow Entry Fields *(Continued)*

Field	Description
End time of flow	System up time, in seconds, at the end of the flow
Destination prefix	Destination IP address prefix
Dest. mask length	Destination address network mask length
Padding	Bytes available to ensure a minimum packet length
Destination AS	AS number of the destination address
Output interface	SNMP index value for the output interface where the router forwards flows
Reserved	Empty field reserved for future usage

For more information about version 5 and version 8 packet formats and fields, see **http://www.caida.org**.

Flow Monitoring Version 9 Format Output Fields

IN THIS SECTION

A detailed explanation of active flow monitoring version 9 packet formats and fields is shown as follows:

- Table 9 on page 27
- Figure 9 on page 30
- Table 10 on page 31

- Figure 11 on page 35
- Table 10 on page 31
- Figure 12 on page 36
- Table 14 on page 36
- Figure 13 on page 37
- Table 15 on page 38

The Junos OS supports the version 9 template formats:

Table 9: Flow Monitoring Version 9 Template Formats

Template	Fields
IPv4	Flow selectors: • Source and destination IP address • Source and destination address prefix mask lengths • Source and destination port numbers • IP protocol and IP type of service • ICMP type Flow nonselectors: • TCP flags • Input and output SNMP • Input bytes • Input packets • Start time • End time

Table 9: Flow Monitoring Version 9 Template Formats *(Continued)*

Template	Fields
MPLS	Flow selectors: • MPLS label 1 • MPLS label 2 • MPLS label 3 Flow nonselectors: • Input and output SNMP • Input bytes • Input packets • Start time • End time
MPLS_IPv4	Flow selectors: • MPLS label 1 • MPLS label 2 • MPLS label 3 • MPLS top-level FEC address Flow nonselectors: • Input and output SNMP • Input bytes • Input packets • Start time • End time

Table 9: Flow Monitoring Version 9 Template Formats *(Continued)*

Template	Fields
IPv6	Flow selectors: • IP protocol and IP type of service • Source and destination port numbers • Input SNMP • Source and destination IPv6 address • ICMP type Flow nonselectors: • Input bytes • Input packets • TCP flags • Output SNMP • Source and destination autonomous system • Last and first switched • IPv6 source and destination mask • IP protocol version • IPv6 next hop

Table 9: Flow Monitoring Version 9 Template Formats *(Continued)*

Template	Fields
Peer AS billing	Flow selectors: • IPv4 class of service • Ingress interface information • BGP peer destination AS number • BGP IPv4 nest hop address Flow nonselectors • Input and output SNMP • Input bytes • Input packets • First switch • Last switched

Figure 9: Version 9 Flow Header Format

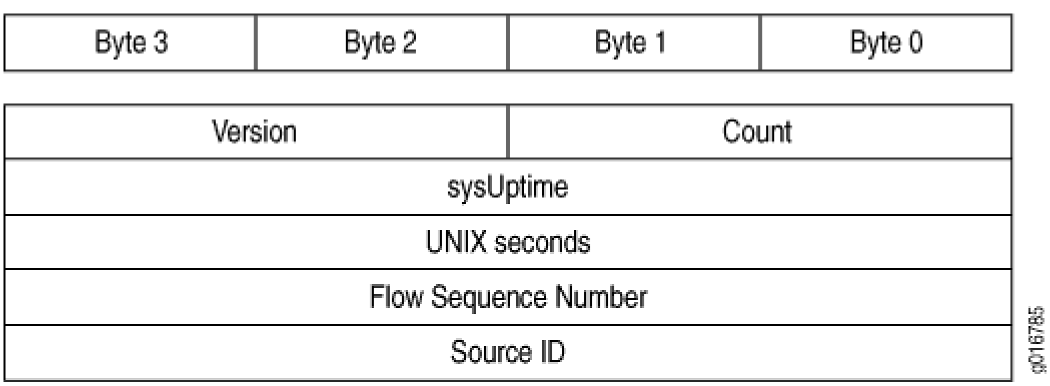


Table 10: Version 9 Flow Header Fields

Field	Description
Version	9
Count	Total number of records in the protocol data unit (PDU) or packet. This number includes all of the options FlowSet records, template FlowSet records, and data FlowSet records.
sysUptime	Current time elapsed, in milliseconds, since the router started.
UNIX seconds	Current seconds since 0000 UTC 1970.
Flow sequence number	Sequence counter of total flows received.
Source ID	32-bit value that identifies the data exporter. Version 9 uses the integrated field diagnostics (IFD) SNMP index of the PIC or device that is exporting the data flow. This field is equivalent to engine type and engine ID fields found in versions 5 and 8.

Figure 10: Version 9 Template FlowSet Format

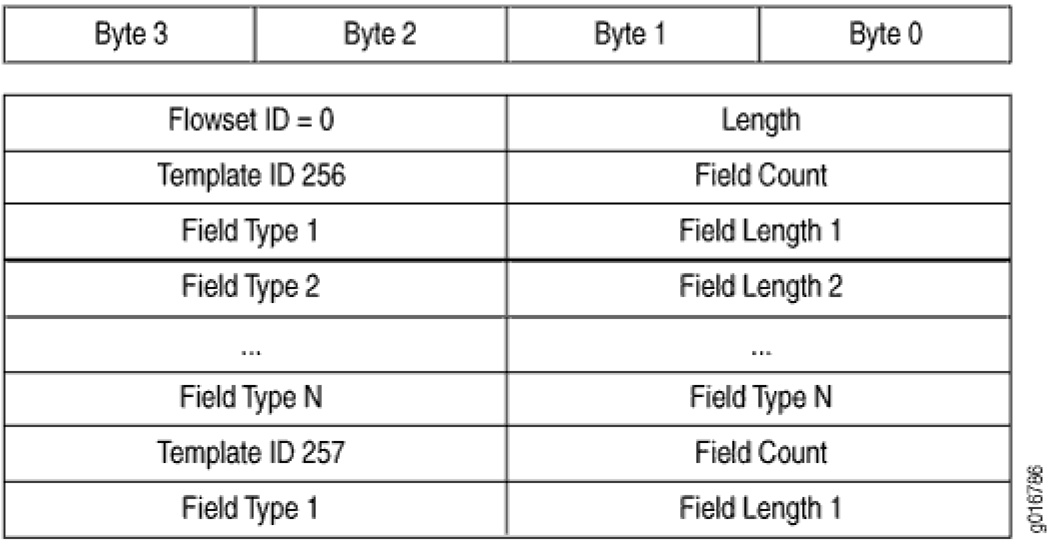


Table 11: Version 9 Template FlowSet Fields

Field	Description
FlowSet ID	FlowSet type. FlowSet ID 0 is reserved for the Template FlowSet.
Length	FlowSet length. Individual template FlowSets might contain multiple template records, which means that the length of template FlowSets varies.
Template ID	Unique template ID assigned to each newly generated template. Templates numbered 256 and higher define data formats. Templates numbered 0 through 255 define FlowSet IDs.
Field Count	Fields in the template record. This field allows the collector to determine the end of the current template record and the start of the next.
Field Type	Field type. These are defined in Table 12 on page 33.

Table 11: Version 9 Template FlowSet Fields *(Continued)*

Field	Description
Field Length	Length, in bytes, of the corresponding field type.

Table 12: Field Type Definitions Supported in Junos OS

Field Type	Description
1	IN_BYTES: The number of bytes associated with an IP flow. By default, the length is 4 bytes.
2	IN_PKTS: The number of packets associated with an IP flow. By default, the length is 4 packets.
4	PROTOCOL: The IP protocol byte.
5	TOS: The type-of-service byte setting of an incoming packet.
6	TCP_FLAGS: The cumulative TCP flags associated with a flow.
7	L4_SRC_PORT: The TCP/UDP source port.
8	IPv4_SRC_ADDR: The IPv4 source address.
9	SRC_MASK: The number of contiguous bits in the source subnet mask.
10	INPUT_SNMP: The IFD SNMP input interface index. By default, the length is 2.
11	L4_DST_PORT: The TCP/UDP destination port number.
12	IPV4_DST_ADDR: The IPv4 destination address.
13	DST_MASK: The number of contiguous bits in the destination subnet mask.

Table 12: Field Type Definitions Supported in Junos OS *(Continued)*

Field Type	Description
14	OUTPUT_SNMP: The IFD SNMP output interface index. By default, the length is 2.
16	SRC_AS: The source autonomous system number. This is always set to zero.
17	DST_AS: The destination autonomous system number. This is always set to zero.
18	BGP_IPV4_NEXT_HOP: The BGP IPV4 next-hop address.
21	LAST_SWITCHED: The uptime of the device (in milliseconds) at which the last packet of the flow was switched.
22	FIRST_SWITCHED: The uptime of the device (in milliseconds) at which the first packet of the flow was switched.
29	IPV6_SRC_MASK: The length of the IPv6 source mask, in contiguous bits.
30	IPV6_DST_MASK: The length of the IPv6 destination mask, in contiguous bits.
32	ICMP_TYPE: The ICMP type.
34	SAMPLING_INTERVAL: The rate at which packets are sampled. As an example, a rate of 100 means that one packet is sampled for every 100 packets in the data flow.
35	SAMPLING_ALGORITHM: The type of algorithm being used. Ox01 indicates deterministic sampling and 0x02 indicates random sampling.
47	MPLS_TOP_LABEL_IP_ADDRESS: The MPLS top- label address.
60	IP_PROTOCOL_VERSION: The IP protocol version being used.
62	IPV6_NEXT_HOP: The IPv6 address of the next-hop router.

Table 12: Field Type Definitions Supported in Junos OS *(Continued)*

Field Type	Description
70	MPLS_LABEL_1: The first MPLS label in the stack.
71	MPLS_LABEL_2: The second MPLS label in the stack.
72	MPLS_LABEL_3: The third MPLS label in the stack.
128	DST_PEER_AS: The destination of the BGP peer AS.

Figure 11: Version 9 Data FlowSet Format

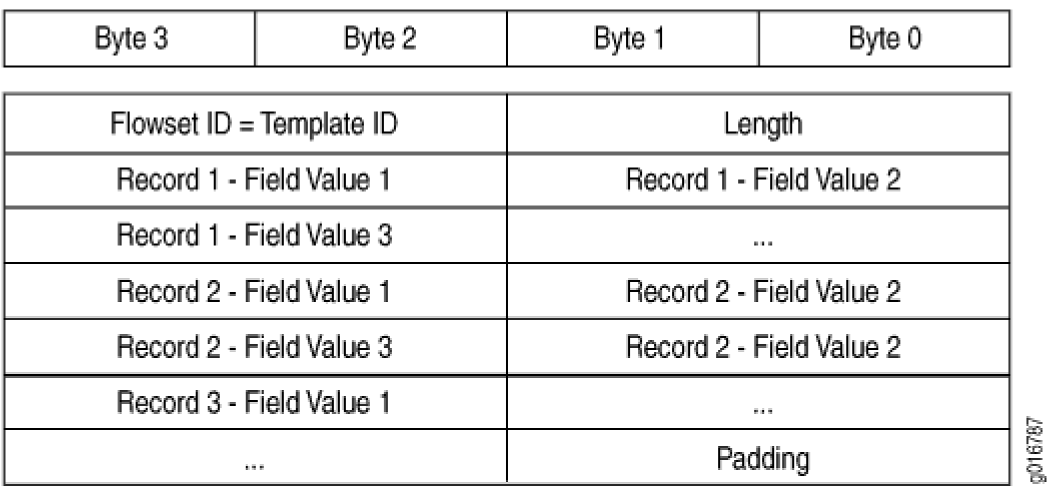


Table 13: Version 9 Data FlowSet Format

Field	Description
FlowSet ID = Template ID	Data FlowSet that associated with a FlowSet ID. The FlowSet ID maps to a previously generated template ID. The flow collector must use the FlowSet ID to find the corresponding template record and decode the flow records from the FlowSet.

Table 13: Version 9 Data FlowSet Format *(Continued)*

Field	Description
Length	FlowSet length. Data FlowSets are fixed in length.
Record Number - Field Value Number	Flow data records, each containing a set of field values. The template record identified by the FlowSet ID dictates the type and length of the field values.
Padding	Bytes (in zeros) that the exporter inserts so that the subsequent FlowSet starts at a 4-byte aligned boundary.

Figure 12: Version 9 Options Template Format

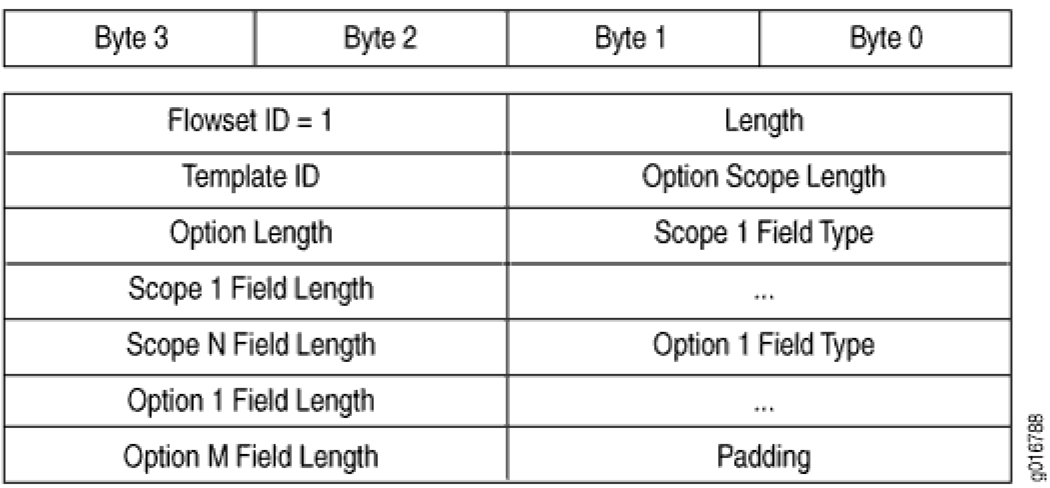


Table 14: Version 9 Options Template Format

Field	Description
FlowSet ID	FlowSet type. FlowSet ID 1 is reserved for the options template.
Length	FlowSet length. Option template FlowSets are fixed in length.

Table 14: Version 9 Options Template Format *(Continued)*

Field	Description
Template ID	Template ID of the options template. Options template values are greater than 255.
Option Scope Length	Length, in bytes, of any scope field definition that is part of the options template record.
Scope 1 Field Type	Relevant process. The Junos OS supports the system process (1).
Scope 1 Field Length	Length, in bytes, of the option field.
Padding	Bytes the exporter inserts so that the subsequent FlowSet starts at a 4-byte aligned boundary.

Figure 13: Active Flow Monitoring Version 9 Options Data Record Format

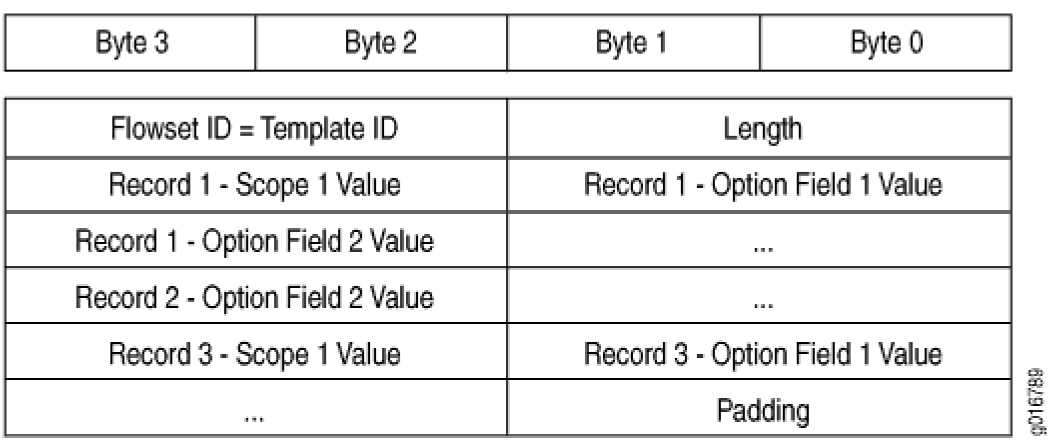


Table 15: Active Flow Monitoring Version 9 Options Data Record Format

Field	Description
FlowSet ID = Template ID	ID that precedes each options data flow record. The FlowSet ID maps to a previously generated template ID. The collector must use the FlowSet ID to find the corresponding template record and decode the options data flow records from the FlowSet.
Length	FlowSet length. Option FlowSets are fixed in length.
Number of Flow Data Records	Remainder of the options data FlowSet is a collection of flow data records, each containing a set of field values. The template record identified by the FlowSet ID dictates the type and length of the field values.
Padding	Bytes (in zeros) the exporter inserts so that the subsequent FlowSet starts at a 4-byte aligned boundary.

IPFIX (Version 10) IPv4 Fields

Field Name	Flow Key	Element ID	Length in Bytes
IPV4_SADDR	Y	8	4
IPV4_DADDR	Y	12	4
IPV4_TOS	Y	5	1
IPV4_PROTO	Y	4	1
TCP_UDP_SPORT	Y	7	2
TCP_UDP_DPORT	Y	11	2

(Continued)

Field Name	Flow Key	Element ID	Length in Bytes
IMCP_TYPE_CODE_IPV4	Y	32	2
IIF	Y	10	4
VLAN_ID	Configurable	58	2
IPV4_SMASK	N	9	1
IPV4_DMASK	N	13	1
SRC_AS	N	16	4
DST_AS	N	17	4
IPV4_NEXTHOP	N	15	4
TCP_FLAGS	N	6	1
OIF	N	14	4
FLOW_BYTES	N	1	8
FLOW_PACKETS	N	2	8
MIN_TTL	N	52	1
MAX_TTL	N	53	1
START_TIME	N	152	8

(Continued)

Field Name	Flow Key	Element ID	Length in Bytes
END_TIME	N	153	8
FIRST_SWITCHED	N	22	4
LAST_SWITCHED	N	21	4
FLOW_END_REASON	N	136	1
IP_PROTOCOL_VERSION	N	60	1
BGP_NEXTHOP_ID	N	18	4
FLOW_DIRECTION	Configurable	61	1
DOT_1Q_VLAN_ID	N	243	2
DOt_1Q_CUSTOMER_VLAN_ID	N	245	2
IP IDENTIFIER	N	54	4

CHAPTER 2

Monitoring Traffic Using Active Flow Monitoring

IN THIS CHAPTER

Configuring Active Flow Monitoring

In active flow monitoring, the router participates in both the monitoring application and in the normal routing functionality of the network. Although the Monitoring Services PIC was designed initially for use as an offline passive flow monitoring tool, it can also be used in an active flow monitoring topology.

Table 16 on page 42 shows which Juniper Networks PICs and corresponding routers support active flow monitoring. For more information on Juniper Networks PICs, see the PIC guide that corresponds to your router.

Table 16: Passive and Active Flow Monitoring PIC Support

PIC Type and Service	M5/M10	M7i/M10i	M20	M40e	M120	M160	T Series/ M320	TX Matrix
Monitoring Services PIC: active flow monitoring	Yes (version 8 only)	Yes	Yes	Yes	No	Yes (version 8 only)	No	No
Monitoring Services II PIC: flow collection services	No	No	No	Yes	No	Yes (version 8 only)	No	No
Adaptive Services PIC: active flow monitoring	Yes (version 8 only)	Yes	Yes	Yes	No	Yes (version 8 only)	No	No

Table 16: Passive and Active Flow Monitoring PIC Support ***(Continued)***

PIC Type and Service	M5/M10	M7i/M10i	M20	M40e	M120	M160	T Series/ M320	TX Matrix
Adaptive Services II PIC: active flow monitoring	Yes (version 8 only)	Yes	Yes	Yes	Yes	Yes (version 8 only)	Yes	Yes
Adaptive Services II PIC: flow-tap services	No	Yes	Yes	Yes	Yes	No	Yes	No
MultiServices 100 PIC: active flow monitoring	No	Yes	No	Yes	No	No	Yes	Yes
MultiServices 400 PIC: active flow monitoring	No	No	No	Yes	Yes	No	Yes	Yes
MultiServices 500 PIC: active flow monitoring	No	No	No	Yes	Yes	No	Yes	Yes
Junos OS-enabled active flow monitoring	No	No	No	No	No	No	No	No

Specified packets can be filtered and sent to the monitoring interface. For the Monitoring Services PIC, the interface name contains the **mo-** prefix. For the Adaptive Services PICs and MultiServices PICs, the interface name contains the **sp-** prefix.

NOTE: If you upgrade from the Monitoring Services PIC to the Adaptive Services PIC or MultiServices PIC for active flow monitoring, you must modify the interface name of your monitoring interface from **mo-***fpc/pic/port* to **sp-***fpc/pic/port*.

The major active flow monitoring actions you can configure at the `[edit forwarding-options]` hierarchy level are as follows:

- Sampling, with the [**edit forwarding-options sampling**] hierarchy. This option extracts limited information (such as the source and destination IP address) from a copy of some of the packets in a flow, while the original packets are forwarded to the intended destination. This option is extended to define active sampling on a per Packet Forwarding Engine basis by defining a sampling instance that specifies a name for the sampling parameters and binding the instance to the particular Packet Forwarding Engine.

- Templates, with the [**edit forwarding-options sampling**] and [**edit services monitoring**] hierarchies. With active flow monitoring support for version 5, version 8, and the customizing version 9, you can use templates to organize the data gathered from sampling.

- Discard accounting, with the [**edit forwarding-options accounting**] hierarchy. This option quarantines unwanted packets, creates flow monitoring records that describe the packets, and discards the packets instead of forwarding them.

- *Port mirroring*, with the [**edit forwarding-options port-mirroring**] hierarchy. This option makes one full copy of all packets in a flow and delivers the copy to a single destination.

- Multiple port mirroring, with the [**edit forwarding-options next-hop-group**] hierarchy. This option allows multiple copies of selected traffic to be delivered to multiple destinations. (Multiple port mirroring requires a Tunnel Services PIC.)

- Flow-tap services processing, with the [**edit services flow-tap**] hierarchy. This option sends copies of packets that match dynamic filter criteria to one or more content destinations.

Unlike passive flow monitoring, you do not need to configure a monitoring group. Instead, you can send filtered packets to a monitoring services or adaptive services interface (**mo-** or **sp-)** by using sampling or discard accounting. Optionally, you can configure port mirroring or multiple port mirroring to direct packets to additional interfaces.

These active flow monitoring options provide a wide variety of actions that can be performed on network traffic flows. However, the following restrictions apply:

- The router can perform either sampling *or* port mirroring at any one time.

- The router can perform either forwarding *or* discard accounting at any one time.

Because the Monitoring Services PIC, Adaptive Services PIC, and MultiServices PIC allow only one action to be performed at any one time, the following configuration options are available:

- Sampling and forwarding

- Sampling and discard accounting

- Port mirroring and forwarding

- Port mirroring and discard accounting
- Sampling and port mirroring on different sets of traffic

To configure active flow monitoring, complete these steps:

- "Defining a Firewall Filter on M, MX and T Series Routers to Select Traffic for Active Flow Monitoring" on page 94
- "Configuring Actively Monitored Interfaces on M, MX and T Series Routers " on page 89
- "Processing IPv4 traffic on an M, MX or T Series Router Using Monitoring services, Adaptive services or Multiservices Interfaces" on page 96
- "Collecting Flow Records" on page 89

Active Flow Monitoring System Requirements

To implement active flow monitoring, your system must meet these minimum requirements:

- Junos 10.4 or later for peer AS billing support on flow monitoring version 9
- Junos 9.3R2 or later for IPv6 support on flow monitoring version 9
- Junos 9.3R2 or later for multiple flows for flow monitoring version 9
- Junos OS Release 9.0 or later for version 9 flow aggregation to multiple flow servers
- Junos OS Release 8.5 or later for active flow monitoring support on MultiServices 500 PICs
- Junos OS Release 8.3 or later for flow monitoring version 9 support, MPLS support, and active flow monitoring support on MultiServices 100 and 400 PICs
- Junos OS Release 8.2 or later for M120 router support and for flow monitoring version 5 and 8 support on MultiServices 100 and 400 PICs
- Junos OS Release 8.1 or later for the flow-tap services application on Adaptive Services II PICs installed in M7i, M10i, M20, M40e, M320, and T Series routers
- Junos OS Release 7.4 or later for port mirroring of IPv6 packets
- Junos OS Release 7.3 or later for active flow monitoring on Adaptive Services II PICs installed in TX Matrix platforms
- Junos OS Release 7.0 or later for active flow monitoring on Adaptive Services II PICs installed in T Series and M320 routers

- Junos OS Release 6.0 or later for the Adaptive Services PIC
- Junos OS Release 5.7 or later for the automatic insertion of AS numbers and SNMP index values for input and output interfaces into records, port mirroring to multiple ports, and discard accounting
- Junos OS Release 5.6 or later for the Monitoring Services PIC
- M5, M7i, M10, M10i, M20, M40e, M120, M160, M320, or T Series router with an Internet Processor II ASIC or later
- Type 1 enhanced FPCs
- Two M Series or T Series PICs of your choice: One to receive incoming traffic and one to forward outgoing traffic (the second PIC or PIM is not necessary for discard accounting)
- Export PICs to connect to the collector or packet analyzer
- Tunnel Services PIC (required for multiple port mirroring or **mo-** interface load balancing)
- Flow collector version 5, 8, or 9
- ES PIC and packet analyzers (optional)

RELATED DOCUMENTATION

Passive Flow Monitoring System Requirements for MX Series Routers | 152
Active Flow Monitoring PIC Specifications | 48

Active Flow Monitoring Applications

Flow monitoring can be used for many different reasons such as network planning, accounting, usage-based network billing, security, and monitoring for Denial-of-Service attacks.

Some examples of the types of things you can use flow monitoring for are:

- Tracking what kind of traffic is entering or exiting an ISP or corporate network.
- Tracking traffic flows between BGP autonomous systems.
- Tracking traffic flows between enterprise network regions.
- Taking a snapshot of the existing quality-of-service (QoS) policy results prior to making changes in QoS policy in case you need to roll back changes later in the process.

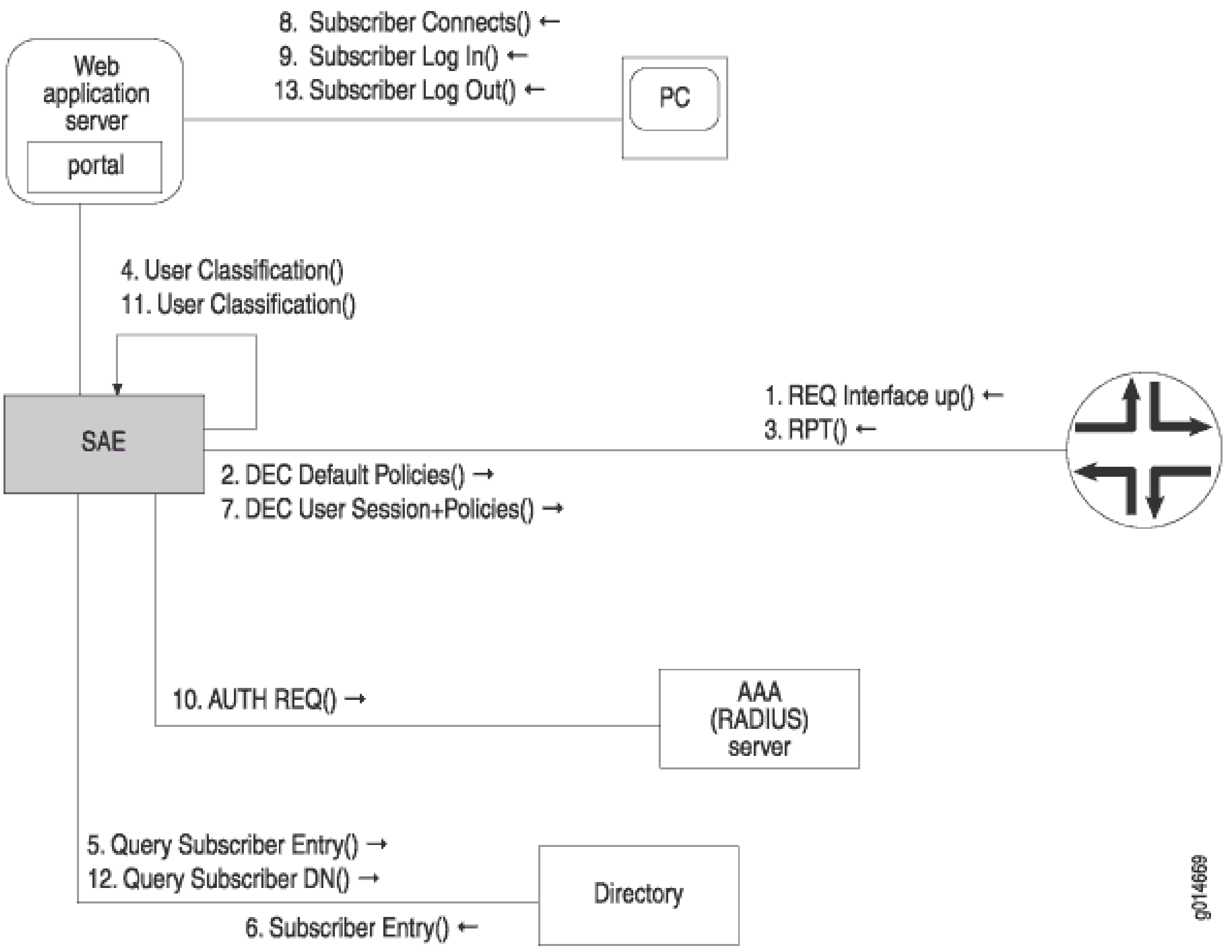
- Verifying that load balancing techniques are performing as intended.
- Capturing a base line of current network performance prior to making changes intended to improve performance so that you know if the changes are helping.
- Discovering if network users at an enterprise are using bandwidth for work-related activities or for non work-related activities.

Examples of how flow monitoring helps with network administration include the following:

- A large service provider uses active flow monitoring on its core uplinks as a way to collect data on the protocols in use, packet sizes, and flow durations to better understand the usage of its Internet service offering. This helps the provider understand where network growth is coming from.
- Service providers bill customers for the data sent or bandwidth used by sending captured flow data to third-party billing software.
- At a large enterprise, VoIP users at aremote site complained of poor voice quality. The flow monitoring reports showed that the VoIP traffic did not have the correct type of service settings.
- Users on an enterprise network, reported network slowdowns. The flow monitoring reports showed that one user's PC was generating a large portion of the network traffic. The PC was infected with malware.
- A growing enterprise planned to deploy new business management software and needed to know what type of network bandwidth demand the new software would create. During the software trial period, flow monitoring reports were used to identify the expected increase in traffic.

Thus, while flow monitoring is traditionally associated with traffic analysis, it also has a role in accounting and security.

Figure 14: Active Flow Monitoring



RELATED DOCUMENTATION

Flow Monitoring Overview

Active Flow Monitoring Overview | 53

Active Flow Monitoring PIC Specifications

For Monitoring Services PIC specifications, see Table 17 on page 49 and Table 18 on page 49. For Adaptive Services PIC specifications, see Table 19 on page 50. For MultiServices PIC specifications, see Table 20 on page 51 and Table 21 on page 52.

Table 17: Monitoring Services PIC Specifications

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	DB-9 diagnostic serial console port
Status LED	One tricolor: • Off—The PIC is offline; it is safe to remove it from the chassis. • Green—The PIC is operating normally. • Amber—The PIC is initializing. • Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: • Off—The service is not running. • Green—The service is running under acceptable load. • Amber—The service is overloaded.

Table 18: Monitoring Services II PIC Specifications

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	N/A

Table 18: Monitoring Services II PIC Specifications *(Continued)*

Specification	Description
Status LED	One tricolor: • Off—The PIC is offline; it is safe to remove it from the chassis. • Green—The PIC is operating normally. • Amber—The PIC is initializing. • Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: • Off—The flow collector is not running. • Green—The flow collector is running under acceptable load. • Amber—The flow collector is overloaded.

Table 19: Adaptive Services PIC Specifications

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	N/A
Status LED	One tricolor: • Off—The PIC is offline; it is safe to remove it from the chassis. • Green—The PIC is operating normally. • Amber—The PIC is initializing. • Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.

Table 19: Adaptive Services PIC Specifications *(Continued)*

Specification	Description
Application LED	One tricolor: • Off—The flow collector is not running. • Green—The flow collector is running under acceptable load. • Amber—The flow collector is overloaded.

Table 20: MultiServices 100 PIC

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	N/A
Status LED	One tricolor: • Off—The PIC is offline; it is safe to remove it from the chassis. • Green—The PIC is operating normally. • Amber—The PIC is initializing. • Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: • Off—The service is not running. • Green—The service is running under acceptable load. • Amber—The service is overloaded.

Table 21: MultiServices 400 PIC

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	N/A
Status LED	One tricolor: • Off—The PIC is offline; it is safe to remove it from the chassis. • Green—The PIC is operating normally. • Amber—The PIC is initializing. • Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: • Off—The service is not running. • Green—The service is running under acceptable load. • Amber—The service is overloaded.

Table 22: MultiServices 500 PIC

Specification	Description
Physical dimensions	Single-wide PIC that occupies one PIC slot
Connectors	N/A

Table 22: MultiServices 500 PIC *(Continued)*

Specification	Description
Status LED	One tricolor: • Off—The PIC is offline; it is safe to remove it from the chassis. • Green—The PIC is operating normally. • Amber—The PIC is initializing. • Red—The PIC has an error or failure; no further harm can be done by removing it from the chassis.
Application LED	One tricolor: • Off—The service is not running. • Green—The service is running under acceptable load. • Amber—The service is overloaded.

RELATED DOCUMENTATION

Passive Flow Monitoring System Requirements for MX Series Routers | 152
Active Flow Monitoring System Requirements | 45

Active Flow Monitoring Overview

Flow monitoring versions 5, 8, and 9 support active flow monitoring. For active flow monitoring, the monitoring station participates in the network as an active router. The major actions the router can perform during active flow monitoring are as follows:

- Sampling—The router selects and analyzes only a portion of the traffic.
- Sampling with templates—The router selects, analyzes, and arranges a portion of the traffic into templates.
- Sampling per sampling instance—The router selects, analyzes, and arranges a portion of the traffic according to the configuration and binding of a sampling instance.

- *Port mirroring*—The router copies entire packets and sends the copies to another interface.
- Multiple port mirroring—The router sends multiple copies of monitored packets to multiple export interfaces with the `next-hop-group` statement at the `[edit forwarding-options]` hierarchy level.
- Discard accounting—The router accounts for selected traffic before discarding it. Such traffic is not forwarded out of the router. Instead, the traffic is quarantined and deleted.
- Flow-tap processing—The router processes requests for active flow monitoring dynamically by using the Dynamic Tasking Control Protocol (DTCP).

RELATED DOCUMENTATION

Flow Monitoring Overview
Understanding Passive Flow Monitoring on MX Series Routers | 154

Active Flow Monitoring Overview

Using a Juniper Networks M Series Multiservice Edge or T Series Core router or EX9200, a selection of PICs (including the Monitoring Services PIC, Adaptive Services [AS] PIC, Multiservices PIC, or Multiservices DPC) and other networking hardware, you can monitor traffic flow and export the monitored traffic. Monitoring traffic allows you to do the following:

- Gather and export detailed information about IP version 4 (IPv4) traffic flows between source and destination nodes in your network.
- Sample all incoming IPv4 traffic on the monitoring interface and present the data in cflowd record format.
- Perform discard accounting on an incoming traffic flow.
- Encrypt or tunnel outgoing cflowd records, intercepted IPv4 traffic, or both.
- Direct filtered traffic to different packet analyzers and present the data in its original format (port mirror).

NOTE: Monitoring Services PICs, AS PICs, and Multiservices PICs must be mounted on an Enhanced Flexible PIC Concentrator (FPC) in an M Series or T Series router.

Multiservices DPCs installed in Juniper Networks MX Series 3D Universal Edge routers support the same functionality, with the exception of the passive monitoring and flow-tap features.

Although the Monitoring Services PIC was designed initially for use as an offline passive flow monitoring tool, it can also be used in an active flow monitoring topology. In contrast, the AS or Multiservices PIC is designed exclusively for active flow monitoring. To use either the Monitoring Services PIC, AS PIC, or Multiservices PIC for active flow monitoring, you must install the PIC in an M Series or T Series router. The router participates in both the monitoring application and in the normal routing functionality of the network.

Starting with Junos OS Release 11.4, support for active monitoring is extended to logical systems running on T Series and MX Series routers. A logical system is a partition created from a physical router that performs independent routing tasks. Several logical systems in a single router with their own interfaces, policies, instances, and routing tables can perform functions handled by several different routers. A shared services PIC handles flows from all the logical systems. Only version 9 flows, IPv4, and MPLS templates are supported. See "Example: Configuring Active Monitoring on an M, MX or T Series Router's Logical System " on page 58 for a sample configuration that enables active monitoring on a logical system.

Specified packets can be filtered and sent to the monitoring interface. For the Monitoring Services PIC, the interface name contains the `mo-` prefix. For the AS or Multiservices PIC, the interface name contains the `sp-` prefix.

NOTE: If you upgrade from the Monitoring Services PIC to the Adaptive Services or Multiservices PIC for active flow monitoring, you must change the name of your monitoring interface from mo-*fpc/pic/port* to sp-*fpc/pic/port*.

The major active flow monitoring actions you can configure at the `[edit forwarding-options]` hierarchy level are as follows:

- Sampling, with the `[edit forwarding-options sampling]` hierarchy. This option sends a copy of the traffic stream to an AS or Monitoring Services PIC, which extracts limited information (such as the source and destination IP address) from some of the packets in a flow. The original packets are forwarded to the intended destination as usual.

- Discard accounting, with the `[edit forwarding-options accounting]` hierarchy. This option quarantines unwanted packets, creates cflowd records that describe the packets, and discards the packets instead of forwarding them.

- *Port mirroring*, with the `[edit forwarding-options port-mirroring]` hierarchy. This option makes one full copy of all packets in a flow and delivers the copy to a single destination. The original packets are forwarded to the intended destination.

- Multiple port mirroring, with the `[edit forwarding-options next-hop-group]` hierarchy. This option allows multiple copies of selected traffic to be delivered to multiple destinations. (Multiple port mirroring requires a Tunnel Services PIC.)

Unlike passive flow monitoring, you do not need to configure a monitoring group. Instead, you can send filtered packets to a monitoring services or adaptive services interface (`mo-` or `sp-`) by using sampling or discard accounting. Optionally, you can configure port mirroring or multiple port mirroring to direct packets to additional interfaces.

These active flow monitoring options provide a wide variety of actions that can be performed on network traffic flows. However, the following restrictions apply:

- The router or switch can perform sampling *or* port mirroring at any one time.

- The router or switch can perform forwarding *or* discard accounting at any one time.

Because the Monitoring Services, AS, and Multiservices PICs allow only one action to be performed at any one time, the following configuration options are available:

- Sampling and forwarding

- Sampling and discard accounting

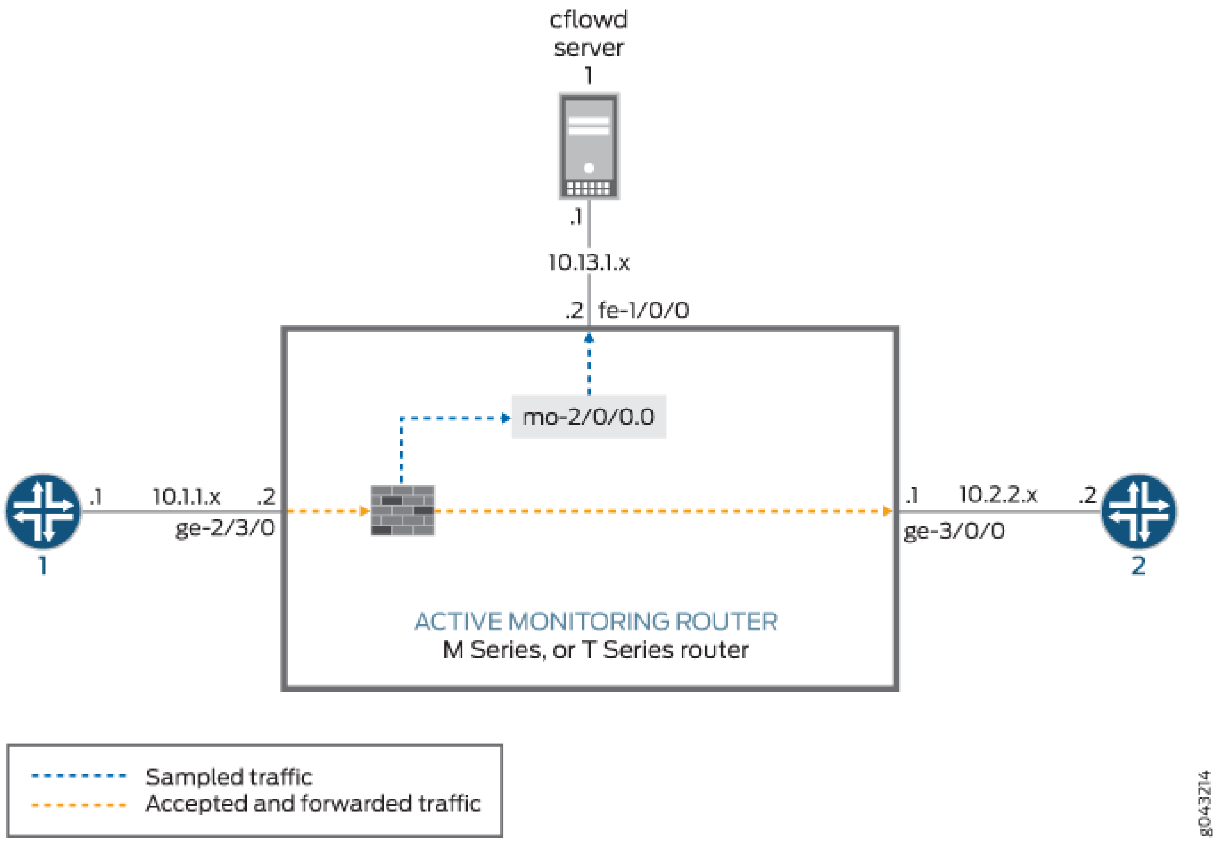
- Port mirroring and forwarding

- Port mirroring and discard accounting

- Sampling and port mirroring on different sets of traffic

Figure 15 on page 57 shows a sample topology.

Figure 15: Active Monitoring Configuration Topology



In Figure 15 on page 57, traffic from Router 1 arrives on the monitoring router's Gigabit Ethernet ge-2/3/0 interface. The exit interface on the monitoring router leading to destination Router 2 is ge-3/0/0, but this can be any interface type (such as SONET, Gigabit Ethernet, and so on). The export interface leading to the cflowd server is fe-1/0/0.

To enable active monitoring, configure a *firewall filter* on the interface ge-2/3/0 with the following match conditions:

- Traffic matching certain firewall conditions is sent to the Monitoring Services PIC using filter-based forwarding. This traffic is quarantined and not forwarded to other routers.
- All other traffic is port-mirrored to the Monitoring Services PIC. Port mirroring copies each packet and sends the copies to the port-mirroring next hop (in this case, a Monitoring Services PIC). The original packets are forwarded out of the router as usual.

RELATED DOCUMENTATION

Example: Configure Active Monitoring on a Logical System for MX Series and PTX Series Routers

This example shows a sample configuration for active monitoring on a logical system running on an MX Series or a PTX Series router.

The following section shows the configuration on the primary router:

```
[edit forwarding-options]
sampling {
    instance inst1 {
        input {
            rate 1;
        }
        family inet;
            output {
                flow-server 198.51.100.2 {
                    port 2055;
                    version9 {
                        template {
                            ipv4;
                        }
                    }
                }
            }
            interface sp-0/1/0 {
                source-address 10.11.12.13;
            }
        }
    }
    family mpls;
        output {
```

```
            flow-server 198.51.100.2 {
                port 2055;
                version9 {
                    template {
                        mpls;
                    }
                }
            }
        }
        interface sp-0/1/0 {
            source-address 10.11.12.13;
        }
    }
}
services {
    flow-monitoring {
        version9 {
            template ipv4 {
                flow-active-timeout 60;
                flow-inactive-timeout 60;
                ipv4-template;
                template-refresh-rate {
                    packets 1000;
                    seconds 10;
                }
                option-refresh-rate {
                    packets 1000;
                    seconds 10;
                }
            }
            template mpls {
                mpls-template;
            }
        }
    }
}
```

The configuration for the logical system uses the input parameters and the output interface for sampling from the primary router. Each logical system should have separate template definitions for the `flow-server` configuration statement. The following section shows the configuration on the logical system:

```
logical-systems {
    ls-1 {
        firewall {
            family inet {
                filter test-sample {
                    term term-1 {
                        then {
                            sample;
                            accept;
                        }
                    }
                }
            }
        }
        interfaces {
            ge-0/0/1 {
                unit 0 {
                    family inet {
                        filter {
                            input test-sample;
                            output test-sample;
                        }
                    }
                }
            }
        }
        forwarding-options {
            sampling {
                instance sample-inst1 {
                    family inet;
                        output {
                            flow-server 198.51.100.2 {
                                port 2055;
                                version9 {
                                    template {
                                        ipv4-ls1;
                                    }
                                }
```

```
                        }
                    }
                }
            }
            family mpls;
                output {
                    flow-server 198.51.100.2 {
                        port 2055;
                        version9 {
                            template {
                                mpls-ls1;
                            }
                        }
                    }
                }
            }
        }
    }
    services {
        flow-monitoring {
            version9 {
                template ipv4-ls1 {
                    flow-active-timeout 60;
                    flow-inactive-timeout 60;
                    ipv4-template;
                    template-refresh-rate {
                        packets 1000;
                        seconds 10;
                    }
                    option-refresh-rate {
                        packets 1000;
                        seconds 10;
                    }
                }
                template mpls-ls1 {
                    mpls-template;
                }
            }
        }
    }
  }
}
```

RELATED DOCUMENTATION

Example: Configuring Flow Monitoring on an MX Series Router with MS-MIC and MS-MPC

IN THIS SECTION

This example shows how you can configure Junos Traffic Vision for flow monitoring on an MX Series Router with MS-MIC and MS-MPC, and contains the following sections:

Configuring Flow Monitoring on MS-MIC

CLI Quick Configuration

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

NOTE: You can follow the same procedure and use the same configuration for configuring flow monitoring on MS-MPC.

Enabling the Services Interface Card

```
set interfaces ms-2/0/0 unit 0 family inet
```

Configuring the Template and Timers

```
set services flow-monitoring version9 template template1
set services flow-monitoring version9 template template1 flow-active-timeout 120
set services flow-monitoring version9 template template1 flow-inactive-timeout 60
set services flow-monitoring version9 template template1 ipv4-template
set services flow-monitoring version9 template template1 template-refresh-rate packets 100
set services flow-monitoring version9 template template1 template-refresh-rate seconds 600
set services flow-monitoring version9 template template1 option-refresh-rate packets 100
set services flow-monitoring version9 template template1 option-refresh-rate seconds 600
```

Configuring Service Set Properties

```
set services service-set ss1 jflow-rules sampling
set services service-set ss1 sampling-service service-interface ms-2/0/0.0
```

Configuring Forwarding Options and Flow Server Settings

```
set forwarding-options sampling input rate 10
set forwarding-options sampling input run-length 18
set forwarding-options sampling family inet output flow-server 10.44.4.3 port 1055
set forwarding-options sampling family inet output flow-server 10.44.4.3 version9 template
template1
set forwarding-options sampling family inet output interface ms-2/0/0.0 source-address
203.0.113.1
```

Step-by-Step Procedure

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

NOTE: The MS interface must be configured with the family type that the collector will be reachable by. If the collector for the sampling traffic is reachable via IPv4, you must set the family inet under the MS interface even if you are only sampling IPv6 and MPLS traffic, for example.

1. Configure the services interface.

```
[edit interfaces]
user@router1# set interfaces ms-2/0/0 unit 0 family inet
user@router1# set interfaces ms-2/0/0 unit 1 family inet6
user@router1# set interfaces ms-2/0/0 unit 2 family mpls
```

2. Configure the template properties and the export policy timers.

```
[edit services]
user@router1# set flow-monitoring version9 template template1
user@router1# set flow-monitoring version9 template template1 flow-active-timeout 120
user@router1# set flow-monitoring version9 template template1 flow-inactive-timeout 60
user@router1# set flow-monitoring version9 template template1 ipv4-template
user@router1# set flow-monitoring version9 template template1 template-refresh-rate packets
100
user@router1# set flow-monitoring version9 template template1 template-refresh-rate seconds
600
user@router1# set flow-monitoring version9 template template1 option-refresh-rate packets 100
user@router1# set flow-monitoring version9 template template1 option-refresh-rate seconds 600
```

Table 23: Quick Reference to Key Configuration Statements at This Hierarchy Level

Configuration Statement	Description
`flow-active-timeout`	Configures the interval (in seconds) after which an active flow is exported. Range is 10 through 600 seconds, and the default value is 60 seconds.
`flow-inactive-timeout`	Configures the interval (in seconds) of inactivity after which a flow is marked inactive. Range is 10 through 600 seconds, and the default value is 60 seconds.
ipv4-template | ipv6-template | mpls-template | mpls-ipv4-template	Specifies the type of traffic for which the template is used for.

Table 23: Quick Reference to Key Configuration Statements at This Hierarchy Level *(Continued)*

Configuration Statement	Description
`template-refresh-rate`	Specifies the template refresh rate either as number of packets (range is 1 through 480,000 and the default value is 4800) or in seconds (the range is 10 through 600 and the default is 600). Because the communication between the flow generator and the flow collector is a one-way communication, the flow generator has to regularly send updates about template definitions to the flow collector. The value configured for this statement controls the frequency of such updates.
`option-refresh-rate`	Specifies the option refresh rate either as number of packets (range is 1 through 480,000 and the default value is 4800) or in seconds (the range is 10 through 600 and the default is 60).

3. Configure service set properties.

```
[edit services]
user@router1# set service-set ss1 jflow-rules sampling
user@router1# set service-set ss1 sampling-service service-interface ms-2/0/0.0
```

Table 24: Quick Reference to Configuration Statements at This Hierarchy Level

Configuration Statement	Description
`sampling`	Configures the service set to handle sampling/flow monitoring activities.
`service-interface`	Specifies the service interface associated with the service set. The interface configured here should match the interface configured at the `[edit forwarding-options sampling family inet output]`. Also, note that the interface should not be associated with any other service set.

4. Configure forwarding options and flow-server properties.

```
[edit forwarding-options]
user@router1# set sampling input rate 10
```

```
user@router1# set sampling input run-length 18
user@router1# set sampling family inet output flow-server 10.44.4.3 port 1055
user@router1# set sampling family inet output flow-server 10.44.4.3 version9 template
template1
user@router1# set sampling family inet output interface ms-2/0/0.0 source-address 203.0.113.1
```

NOTE: You can specify the sampling parameters either at the global level (as shown in this example) or at the FPC level by defining a sampling instance. To define a sampling instance, include the `instance` statement at the `[edit forwarding-options sampling]` hierarchy level, and the `sampling-instance` statement at the `[edit chassis fpc number]` hierarchy level to associate the sampling instance with an FPC. Under the `[edit forwarding-options sampling instance instance]` hierarchy level, you must also include the `input` and `output` configurations explained in this step.

Table 25: Quick Reference to Key Configuration Statements at this Hierarchy Level

Configuration Statement	Description
`rate`	The ratio of the number of packets to be sampled. For example, if you specify a rate of 10, every tenth packet (1 packet out of 10) is sampled. The range is 1 through 16000000(16M).
`run-length`	The number of samples following the initial trigger event. This enables you to sample packets following those already being sampled. The range is 0 through 20, and the default is 0.
`flow-server`	A host system to collect sampled flows using the version 9 format.
`source-address`	An IPv4 address to be used as the source address of the exported packet.

Result

From the configuration mode, confirm your configuration by entering the `show chassis fpc 2`, `show interfaces`, and `show forwarding-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@router1# show interfaces
ms-2/0/0 {
    unit 0 {
        family inet;
    }
}
```

```
user@router1# show services
flow-monitoring {
    version9 {
        template template1 {
            flow-active-timeout 120;
            flow-inactive-timeout 60;
            template-refresh-rate {
                packets 100;
                seconds 600;
            }
            option-refresh-rate {
                packets 100;
                seconds 600;
            }
            ipv4-template;
        }
    }
}
    service-set ss1 {
        jflow-rules {
            sampling;
        }
        sampling-service {
            service-interface ms-2/0/0.0
```

```
        }
    }
```

```
user@router1# show forwarding-options
sampling {
    input {
        rate 10;
        run-length 18;
    }
    family inet {
        output {
            flow-server 10.44.4.3 {
                port 1055;
                version9 {
                    template {
                        template1;
                    }
                }
            }
            interface ms-2/0/0.0 {
                source-address 203.0.113.1;
            }
        }
    }
}
```

Hardware and Software Requirements

This example requires an MX Series router that has:

- Junos OS Release 13.2 running on it.
- An MS-MIC installed in it.

Junos Traffic Vision Support on MS-MIC and MS-MPC

Junos Traffic Vision (previously known as Jflow) is the accounting service that is available on the MS-MIC and MS-MPC. Junos Traffic Vision enables users to keep track of the packets received on the MS-MIC or MS-MPC and to generate flow records that contain information such as the source address of the packet, the destination address of the packet, packets and byte counts, and so on. Junos Traffic Vision implementation does not interrupt the traffic, instead it makes a copy of the incoming packet and sends that copy to the service interface card for analyzing the information and maintaining the record.

Starting with Release 13.2, the Junos OS extension-provider packages come preinstalled on a multiservices MIC and MPC (MS-MIC and MS-MPC). The `adaptive-services` configuration at the `[edit chassis fpc *number* pic *number*]` hierarchy level is preconfigured on these cards.

Before you configure Junos Traffic Vision on an MS-MIC or an MS-MPC, you must create a firewall filter that has `sample` configured as action, and apply that to the interface on which you want to monitor the traffic. The flow-collector in Junos Traffic Vision implementations is a device for collecting the flow records. The flow collector is typically deployed outside the network.

> **NOTE**: For more information about configuring firewall filters, see the Junos OS *Firewall Filters Configuration Guide*.

On MS-MIC and MS-MPC, Junos OS supports Junos Traffic Vision Version 9 (v9). Junos Traffic Vision v9 supports sampling of IPv4, IPv6, and MPLS traffic. A services interface card is essential for the v9 implementation, and hence this is often known as PIC-based monitoring.

You can configure the maximum time for which the flow records are stored on the services interface card. The active timeout and inactive timeout values, configured while defining the template, control the export of flow records to the collector. An MS-MIC can store a maximum of 14 million flow records, whereas an MS-MPC can store upto 30 million flows per NPU.

> **NOTE**: In Junos Traffic Vision configurations using the Junos OS extension-provider package, modifying the following statements after flow monitoring has been initiated causes all existing flows to expire:
>
> - At the `[edit forwarding-options sampling instance *instance-name* family (inet |inet6 |mpls) output]` and `[edit forwarding-options sampling family (inet |inet6 |mpls) output]` hierarchy levels:
>   - `flow-server *ip-address*`
>   - `flow-server port *port-number*`
>   - `flow-server template *template*`
> - At the `[edit services flow-monitoring version9 template *template-name* mpls-ipv4-template]` and `[edit services flow-monitoring version9 template *template-name* mpls-template]` hierarchy levels:
>   - `label-position`
>
> Because these changes can disrupt the ongoing flow monitoring, we recommend that you do not change these values after flow monitoring has been initiated on a device. The

changes made to these configuration statements when flow monitoring is going on, apply only to the newly created flows.

Also, note that these changes do not disrupt flow monitoring on devices running Jflow configuration using the Junos OS Layer 2 services package. However, even in the case of Layer 2 service package-based configuration, the changes are applied only to the newly created flows. The existing flows continue to use the initial settings.

NOTE: When Junos Traffic Vision is configured on the MS-MIC and MS-MPC, the next-hop address and outgoing interfaces are incorrectly displayed in the IPv4 and IPv6 flow records when the destination of the sampled flow is reachable through multiple paths.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the Junos Traffic Vision Configuration

Purpose

Verify that Junos Traffic Vision is enabled on the router.

Action

From operational mode, enter the `show services accounting status` command.

```
user@router1> show services accounting status
  Service Accounting interface: ms-2/0/0
  Export format: 9, Route record count: 2093
```

```
  IFL to SNMP index count: 35, AS count: 2
  Configuration set: Yes, Route record set: Yes, IFL SNMP map set: Yes
```

Meaning

Shows the service interface on which monitoring is configured, and also provides information about the export format used (version 9 in this case).

Viewing the Flow Details

Purpose

View the flow details on the interface configured for flow monitoring.

Action

From operational mode, enter the `show services accounting flow` command.

```
user@router1> show services accounting flow
Flow information
    Service Accounting interface: ms-2/0/0, Local interface index: 229
    Flow packets: 220693, Flow bytes: 24276230
    Flow packets 10-second rate: 99, Flow bytes 10-second rate: 10998
    Active flows: 10, Total flows: 12
    Flows exported: 199, Flows packets exported: 718
    Flows inactive timed out: 2, Flows active timed out: 199
```

Viewing Details of Errors That Occurred on the Services Interface

Purpose

View details of errors, if any, on the interface that is configured for flow monitoring.

Action

From operational mode, enter the `show services accounting errors` command.

```
user@router1> show services accounting errors
Error information
```

```
    Service Accounting interface: ms-2/0/0
    Service sets dropped: 0, Active timeout failures: 0
    Export packet failures: 0, Flow creation failures: 0
    Memory overload: No
```

RELATED DOCUMENTATION

Multiservices MIC and Multiservices MPC (MS-MIC and MS-MPC) Overview

Example: Configuring Junos VPN Site Secure on MS-MIC and MS-MPC

Configuring Services Interface Redundancy with Flow Monitoring

Active monitoring services configurations on AS, Multiservices PICs, and Multiservices DPCs support redundancy. To configure redundancy, you specify a redundancy services PIC (`rsp`) interface in which the primary AS or Multiservices PIC is active and a secondary PIC is on standby. If the primary PIC fails, the secondary PIC becomes active, and all service processing is transferred to it. If the primary PIC is restored, it remains on standby and does not preempt the secondary PIC; you need to manually restore the services to the primary PIC. To determine which PIC is currently active, issue the `show interfaces redundancy` command.

NOTE: On flow-monitoring configurations, the only service option supported is *warm standby*, in which one backup PIC supports multiple working PICs. Recovery times are not guaranteed, because the configuration must be completely restored on the backup PIC after a failure is detected. However, configuration is preserved and available on the new active PIC.

As with the other services that support warm standby, you can issue the `request interfaces (revert | switchover)` command to switch manually between the primary and secondary flow monitoring interfaces.

For more information, see *Configuring AS or Multiservices PIC Redundancy*. For information on operational mode commands, see the CLI Explorer.

A sample configuration follows.

```
interface {
    rsp0 {
        redundancy-options {
```

```
            primary sp-0/0/0;
            secondary sp-1/3/0;
        }
        unit 0 {
            family inet;
        }
    }
}
interface {
    ge-0/2/0 {
        unit 0 {
            family inet {
                filter {
                    input as_sample;
                }
            }
            address 10.58.255.49/28;
        }
    }
}
forwarding-options {
    sampling {
        instance instance1 { # named instances of sampling parameters
            input {
                rate 1;
                run-length 0;
                max-packets-per-second 65535;
            }
            family inet {
                output {
                    flow-server 10.10.10.2 {
                        port 5000;
                        version 5;
                    }
                    flow-active-timeout 60;
                    interface rsp0 {
                        source-address 10.10.10.1;
                    }
                }
            }
        }
    }
}
```

```
firewall {
    filter as_sample {
        term t1 {
            then {
                sample;
                accept;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Active Flow Monitoring Overview | 54

Configuring Flow Monitoring | 5

Directing Replicated Flows from M and T Series Routers to Multiple Flow Servers | 658

Example: Configure Active Monitoring on a Logical System for MX Series and PTX Series Routers | 58

Configuring Inline Active Flow Monitoring Using Routers, Switches or NFX250

Inline active flow monitoring is implemented on the Packet Forwarding Engine. The Packet Forwarding Engine performs functions such as creating and updating flows, and updating flow records. The flow records are sent out in industry-standard IPFIX or version 9 format.

On routers with MS-PICs or MS-DPCs, IPv4 and IPv6 fragments are processed accurately. The flow monitoring application creates two flows for every fragmented flow. The first fragment that has the complete Layer 4 information forms the first flow with 5-tuple data and subsequently, all the fragmented packets related to this flow form another flow with the Layer 4 fields set to zero.

The following limitations and restrictions apply to the inline active flow monitoring feature:

- Configuring both sFlow and inline active flow monitoring on the same interface leads to unexpected behavior. Therefore, configure these features on separate interfaces.

- Configuring both egress port mirroring and inline active flow monitoring on the same interface leads to unexpected behavior. Therefore, configure these features on separate interfaces.

- Ingress and egress sampling are sent to the same host-path queue. The packet rate in the queue is shared across ingress and egress sampled packets.
- Forwarding class configuration is not effective. Export record packets are always considered to be control frames and as such are pushed to the network-control queue.
- If multiple inline active flow monitoring firewall filters match to a flow, only the actions of the first filter are taken.
- In ingress sampling, if the destination port is on an aggregated Ethernet interface, the output interface is invalid.

The following considerations apply to the inline active flow monitoring instance configuration:

- Sampling run-length and clip-size are not supported.
- For inline configurations, collectors are not reachable via management interfaces, such as `fxp0`.
- Inline active flow monitoring does not support `cflowd`. Therefore, inline flow monitoring does not support the local dump option, which is available only with cflowd.
- Inline active flow monitoring is not supported when you enable Next Gen Services on an MX Series router.
- The number of collectors that are supported depends on the device:
  - In Junos OS Release 16.2 and in Junos OS Release 16.1R3 and earlier, you can configure only one collector under a family for inline active flow monitoring. Starting with Junos OS Release 16.1R4 and 17.2R1, you can configure up to four collectors under a family for inline active flow monitoring. Starting with Junos OS Evolved 20.3R1, for the PTX10003 and PTX10008 (with the JNP10K-LC1201 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring. Starting with Junos OS Evolved 20.4R1, for the PTX10001-36MR and the PTX10008 (with the JNP10K-LC1202 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring. Starting with Junos OS Evolved 21.1R1, for the PTX10004 router, you can configure up to four collectors for inline active flow monitoring. The Packet Forwarding Engine (PFE) can export the flow record, flow record template, option data, and option data template packet to all configured collectors. To configure a collector under a family for inline active flow monitoring, configure the `flow-server` statement at the `[edit forwarding-options sampling instance` *`instance-name`* `family (inet | inet6 | mpls) output]` hierarchy level. To specify up to four collectors, include up to four `flow-server` statements.
  - For inline configurations on all other devices, each family can support only one collector.

Inline active flow monitoring is configured using statements from four hierarchy levels:

- `[edit chassis]`—At this level, you associate the sampling instance with the FPC on which the media interface is present (except on the MX80 and MX104—see "Configuring Inline Active Flow

Monitoring on MX80 and MX104 Routers" on page 555). If you are configuring sampling of IPv4 flows, IPv6 flows or VPLS flows (Junos OS only), you can configure the flow hash table size for each family, as described below.

- `[edit firewall]`—At this level, you configure a firewall filter for the family of traffic to be sampled. You must attach this filter to the interface on which you want to sample the traffic.

- `[edit forwarding-options]`—At this level, you configure a sampling instance and associate the template with the sampling instance. At this level, you also configure the flow-server IP address and port number as well as the flow export rate.

- `[edit services flow-monitoring]` —At this level, you configure the template properties for inline flow monitoring.

Before you configure inline active flow monitoring, you should ensure that you have adequately-sized hash tables for IPv4, IPv6, MPLS, and VPLS flow sampling. (VPLS flow sampling is Junos OS only). These tables can use one to fifteen 256K areas. Starting with Junos OS Release 16.1R1 and 15.1F2, the IPv4 table is assigned a default value of 1024. Prior to Junos OS Release 16.1 and 15.1F2, the IPv4 table is assigned a default value of fifteen 256K areas. The IPv6 table is assigned a default value of 1024, and the VPLS table is assigned a default value of 1024. When anticipated traffic volume requires larger tables, allocate larger tables.

To allocate flow hash tables:

1. Go to the [edit chassis fpc 0 inline-services flow-table-size] hierarchy level for inline services on the FPC that processes the monitored flows.

```
[edit]
user@host# edit chassis fpc 0 inline-services flow-table-size
```

2. Specify the required sizes for the sampling hash tables.

```
[edit  chassis fpc 0 inline-services flow-table-size]
user@host# set bridge-flow-table-size units
user@host# set ipv4-flow-table-size  units
user@host# set ipv6-flow-table-size units
user@host# set mpls-flow-table-size  units
user@host# set vpls-flow-table-size units
```

NOTE: Starting in Junos OS Release 18.2R1, the `bridge-flow-table-size` option is available and the `vpls-flow-table-size` option is deprecated; use the `bridge-flow-table-size` option instead. The `bridge-flow-table-size` option supports both VPLS and bridge records.

NOTE:

Starting with Junos OS Release 17.3R1, the maximum supported flow table size varies by line card type. See *flow-table-size* for how the size varies by line card type.

Also, starting in Junos OS Release 16.1R1 and 15.1F2, changing the flow hash table size does *not* automatically reboot the FPC (for earlier releases changing the flow hash table size triggers the FPC to reboot).

To configure inline active flow monitoring on MX Series routers (except for MX80 and MX104 routers), EX Series switches, and T4000 routers with Type 5 FPC:

1. Enable inline active flow monitoring and specify the source address for the traffic.

```
[edit forwarding-options sampling instance instance-name family (bridge | inet | inet6 |
mpls | vpls ) output]
user@host# set inline-jflow source address address
```

2. Specify the template to use with the sampling instance.

```
[edit forwarding-options sampling instance instance-name family (bridge | inet | inet6 |
mpls | vpls ) output flow-server hostname]
user@host# set (version9 | version-ipfix) template template-name
```

3. Configure a template to specify output properties.

```
[edit services flow-monitoring]
user@host# set (version-ipfix | version9) template template-name
```

4. (Optional) Configure the interval after which an active flow is exported.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-active-timeout seconds
```

5. (Optional) Configure the interval of activity that marks a flow as inactive.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-inactive-timeout seconds
```

6. (Optional) Configure the template refresh rate in either number of packets or number of seconds.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set template-refresh-rate (packets packets | seconds seconds)
```

7. (Optional) Configure the refresh rate in either number of packets or number of seconds.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set option-refresh-rate (packets packets | seconds seconds)
```

8. Specify the type of record that the template is used for.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set (bridge-template | ipv4-template | ipv6-template | mpls-ipv4-template | mpls-
template | peer-as-billing-template | vpls-template)
```

   The `vpls-template` option is only for IPFIX templates.

   Starting in Junos OS Release 18.2R1, the `bridge-template` option is available and the `vpls-template` option is deprecated; use the `bridge-template` option instead. The `bridge-template` option (Junos OS only) supports both VPLS and bridge records and is for both IPFIX and version9 templates.

   Starting in Junos OS Release 18.4R1, the `mpls-ipv4-template` option is deprecated for inline flow monitoring. To configure MPLS records starting in Junos OS Release 18.4R1, use the `mpls-template` option and the `tunnel-observation` option. This is described in step "9" on page 78.

9. Starting in Junos OS Release 18.4R1 for the MX Series, if you are configuring any type of MPLS flow records, perform the following:

   a. Specify the MPLS template.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set mpls-template
```

b. Configure the type of MPLS flow records to create.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set tunnel-observation [ipv4 | ipv6]
```

The `tunnel-observation` values enable the creation of the following types of flow records:

- `ipv4`—MPLS-IPv4 flows
- `ipv6`—MPLS-IPv6 flows

You can configure multiple values for `tunnel-observation`.

For an MPLS traffic type that does *not* match any of the `tunnel-observation` values, plain MPLS flow records are created. For example, if you only configure `ipv4`, then MPLS-IPv6 traffic results in plain MPLS flow records.

If you do not configure `tunnel-observation`, plain MPLS flow records are created.

c. If you are running inline flow monitoring on a Lookup (LU) card, enable sideband mode to create MPLS-IPv6 flow records.

```
[edit chassis fpc slot-number inline-services]
user@host# set use-extended-flow-memory
```

If you are running inline flow monitoring on an LU card and do not enable sideband mode, then MPLS-IPv6 traffic results in plain MPLS flow records.

10. (Optional) Include the flow direction value in the template.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-key flow-direction
```

The reported data field contains 0x00 (ingress) or 0x01 (egress). If you do not include the `flow-key flow-direction` statement, the flow direction data field contains the invalid value 0xFF.

11. (Optional) Include VLAN IDs in both the ingress and egress directions in the flow key.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-key vlan-id
```

This statement is not required for ingress and egress VLAN ID reporting on interfaces.

12. Associate the sampling instance with the FPC on which you want to implement inline active flow monitoring.

    For MX240, MX480, MX960, MX2010, MX2020, use the following command:

    ```
    [edit ]
    user@host# set chassis fpc fpc-number sampling-instance instance-name
    ```

    a. Confirm the configuration by running the following show command:

       ```
       user@host# show chassis
       ```

       ```
       fpc 0 {
               sampling-instance sample-ins1;
           }
       ```

    For MX5, MX10, MX40, and MX80, use the following command:

    ```
    [edit ]
    user@host# set chassis tfeb slot 0 sampling-instance instance-name
    ```

    a. Confirm the configuration by running the following show command:

       ```
       user@host# show chassis
       ```

       ```
       tfeb {
           slot 0 {
               sampling-instance sample-ins1;
           }
       }
       ```

For MX104, use the following command:

```
[edit ]
user@host# set chassis afeb slot 0 sampling-instance instance-name
```

a. Confirm the configuration by running the following show command:

```
user@host# show chassis
```

```
afeb {
    slot 0 {
        sampling-instance sample-ins1;
    }
}
```

This example shows the sampling configuration for an instance that supports inline active flow monitoring on `family inet`:

```
[edit]
user@host> show forwarding-options
sampling {
    instance {
        sample-ins1 {
            input {
                rate 1;
            }
            family inet {
                output {
                    flow-server 192.0.2.2 {
                        port 2055;
                        version-ipfix {
                            template {
                                ipv4;
                            }
                        }
                    }
                    inline-jflow {
                        source-address 10.11.12.13;
```

```
                    }
                }
            }
        }
    }
}
```

Here is the output format configuration:

```
[edit]
user@host> show services flow-monitoring
services {
    flow-monitoring {
        version-ipfix {
            template ipv4 {
                flow-active-timeout 60;
                flow-inactive-timeout 60;
                ipv4-template;
                template-refresh-rate {
                    packets 1000;
                    seconds 10;
                }
                option-refresh-rate {
                    packets 1000;
                    seconds 10;
                }
            }
        }
    }
}
```

The following example shows the output format configuration for chassis `fpc0`:

```
[edit]
user@host> show services flow-monitoring
sampling-instance instance-1; {
    inline-services {
        flow-table-size {
            ipv4-flow-table-size 8;
            ipv6-flow-table-size 7;
        }
```

```
    }
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
23.4R1-EVO	Starting in Junos OS Evolved 23.4R1 for the ACX7024X, ACX7332, and ACX7348 routers, we support ingress and egress sampling of IPv4 and IPv6 traffic on aggregated Ethernet and IRB interfaces, for both the IPFIX and version 9 export formats. You can configure up to four IPv4 collectors for inline active flow monitoring. You can sample traffic mapped to non-default VRFs, but you cannot export IPFIX or V9 records of those sampled packets through a non-default or management VRF.
23.1R1-EVO	Starting in Junos OS Evolved Release 23.1R1, for the PTX10001-36MR, PTX10003, and PTX10004 routers, as well as the PTX10008 and PTX10016 routers (with the JNP10K-LC1201 or the JNP10K-LC1202 line card and the JNP10008-SF3) routers, we support IPv6 addresses for IPFIX and version 9 collectors. You can configure either IPv4 or IPv6 collectors for each family within a sampling instance; you cannot specify both for the same family. You can specify up to four collectors for each family. You specify the destination server address with the `flow-server` *address* statement and the source address with the `inline-jflow source-address` *address* statement at the `[edit forwarding-options sampling instance` *name* `family (inet | inet6 | mpls) output]` hierarchy level.
23.1R1-EVO	Starting in Junos OS Evolved 23.1R1 for the ACX7100 and ACX7509 routers, we support ingress and egress sampling of IPv4 and IPv6 traffic on aggregated Ethernet and IRB interfaces, for both the IPFIX and version 9 export formats. You can configure up to four IPv4 collectors for inline active flow monitoring. You can sample traffic mapped to non-default VRFs, but you cannot export IPFIX or V9 records of those sampled packets through a non-default or management VRF.
21.4R1-EVO	Starting in Junos OS Evolved Release 21.4R1 for PTX Series, you can export BGP community and AS path information using IP Flow Information Export (IPFIX) information elements 483 through 491, 16, and 17, per RFCs 8549 and 6313. Content providers can use this information to identify a transit service provider degrading the quality of the service. You configure these elements with the statement `data-record-fields` at the `[edit services flow-monitoring version-ipfix template template-name]` hierarchy level.
21.1R1-EVO	Starting with Junos OS Evolved 21.1R1, for the PTX10004 router, you can configure up to four collectors for inline active flow monitoring.

20.4R1-EVO	Starting with Junos OS Evolved 20.4R1, for the PTX10001-36MR and the PTX10008 (with the JNP10K-LC1202 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring.
20.3R1-EVO	Starting with Junos OS Evolved 20.3R1, for the PTX10003 and PTX10008 (with the JNP10K-LC1201 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring.
19.3R2	Inline active flow monitoring is not supported when you enable Next Gen Services on an MX Series router.
18.4R1	Starting in Junos OS Release 18.4R1, the `mpls-ipv4-template` option is deprecated for inline flow monitoring. To configure MPLS records starting in Junos OS Release 18.4R1, use the `mpls-template` option and the `tunnel-observation` option.
18.2R1	Starting in Junos OS Release 20.3R1 for QFX10002-60C switches, you can configure inline active flow monitoring for IPv4 and IPv6 traffic. Both IPFIX and version 9 templates are supported.
18.2R1	Starting in Junos OS Release 18.2R1, the `bridge-flow-table-size` option is available and the `vpls-flow-table-size` option is deprecated; use the `bridge-flow-table-size` option instead.
18.2R1	Starting in Junos OS Release 18.2R1, the `bridge-template` option is available and the `vpls-template` option is deprecated; use the `bridge-template` option instead.
17.2R1	Starting in Junos OS Release 17.2R1 for QFX10002 switches, we added support for inline active flow monitoring with IPFIX templates.
16.1R4	In Junos OS Release 16.2 and in Junos OS Release 16.1R3 and earlier, you can configure only one collector under a family for inline active flow monitoring. Starting with Junos OS Release 16.1R4 and 17.2R1, you can configure up to four collectors under a family for inline active flow monitoring.
16.1R1	Also, starting in Junos OS Release 16.1R1 and 15.1F2, changing the flow hash table size does *not* automatically reboot the FPC.
16.1R1	Starting with Junos OS Release 16.1R1 and 15.1F2, the IPv4 table is assigned a default value of 1024.

RELATED DOCUMENTATION

Configuring Inline Active Flow Monitoring on MX80 and MX104 Routers | 555

Example: Configuring Inline Active Flow Monitoring on MX Series and T4000 Routers | 585

inline-jflow

Configuring Flow Offloading on MX Series Routers

The Junos OS enables you to configure flow offloading for PICS on MX Series routers using Modular Port Concentrator (MPCs) with Modular Interface Cards (MICs). Flows are offloaded to Fast Update Filters (FUFs) on the Packet Forwarding Engine. Offloading produces the greatest benefits when applied to long-lasting or high-bandwidth flows.

The maximum number of active offloads is 200,000 per PIC. When offloaded flows are deleted, more flows can be offloaded.

To configure flow offloading:

- At the `[edit interfaces` *`interface-name`* `services-options]` hierarchy level, enter the `trio-flow-offload minimum-bytes` *`minimum-bytes`* statement.

```
user@host# edit services interface-name
[edit services interface-name services-options]
user@host# set trio-flow-offload minimum-bytes minimum- bytes
```

In the following example, flows are offloaded when they consist of no less than 1024 bytes:

```
user@host# edit services ms-0/1/0
[edit services ms-0/1/0 services-options]
user@host# set trio-flow-offload minimum-bytes 1024
```

RELATED DOCUMENTATION

trio-flow-offload

Configuring Active Flow Monitoring on PTX Series Packet Transport Routers

You can use flow monitoring to help with network administration. Active flow monitoring on PTX Series routers allows you to collect sampled packets, then the router does GRE encapsulation of the packets and sends them to a remote server for flow processing. The GRE encapsulation includes an interface index and GRE key field. The GRE encapsulation removes MPLS tags. You configure one or more port-mirroring instances to define which traffic to sample and configure a server to receive the GRE encapsulated packets. You configure a firewall filter on interfaces where you want to capture flows. You can configure as many as 48 port-mirroring instances.

To configure the router to do GRE encapsulation of sampled packets and send them to a remote server for flow processing:

1. Configure one or more server profiles that specify a host where GRE encapsulated sampled packets are sent, and optionally, a source address to include in the header of each sampled packet.

   a. Specify a name for each server profile and an IP address of the host where sampled packets are sent:

   ```
   [edit services hosted-services]
   user@host# set server-profile server-profile-name server-address ipv4-address
   ```

   b. (Optional) For each server profile, specify a source address to include in the header of each sampled packet:

   ```
   [edit services hosted-services server-profile server-profile-name]
   user@host# set client-address ipv4-address
   ```

   NOTE: The default client address is 0.0.0.0. You must specify an IPv4 address as the client address. You can also specify the loopback address or management interface address as the client address.

2. Configure one or more port-mirroring instances.

   a. Specify a name for each port-mirroring instance:

   ```
   [edit forwarding-options port-mirroring]
   user@host# set instance instance-name
   ```

> **NOTE**: You can configure a maximum of 48 port-mirroring instances.

b. Specify a protocol family for each port-mirroring instance:

```
[edit forwarding-options port-mirroring instance instance-name]
user@host# set family (inet | inet6 )
```

3. To set the ratio of the number of packets to sample, specify a value from 1 through 65,535 for each port-mirroring instance:

```
[edit forwarding-options port-mirroring instance instance-name input]
user@host# set rate number
```

> **NOTE**: You must specify a value for the `rate` statement. The default value is zero, which effectively disables sampling. If, for example, you specify a rate value of 4, every fourth packet (1 packet out of 4) is sampled.

4. (Optional) Specify the number of samples to collect after the initial trigger event for each port-mirroring instance:

```
[edit forwarding-options port-mirroring instance instance-name input]
user@host# set run-length number
```

> **NOTE**: The default value is zero. You can specify a number up to 20.

5. To designate a host where sampled traffic is sent, specify the name of server profile configured at the `[edit services hosted-services]` hierarchy level for each port-mirroring instance:

```
[edit forwarding-options port-mirroring instance instance-name family ( inet | inet6) output]
user@host# set server-profile server-profile-name
```

6. Configure one or more firewall filters.

a. For each firewall filter, specify a protocol family, filter name, and match conditions:

```
[edit firewall]
user@host# set filter family (inet | inet6) filter filter-name term term-name from match-
condtions
```

b. For each firewall filter you configure, specify the name of a port-mirroring instance you configured at the `[edit forwarding-options]` hierarchy level as a nonterminating action so that the traffic that matches that instance is sampled:

```
[edit firewall family (inet | inet6) filter filter-name term term-name]
user@host# set then port-mirroring instance instance-name
```

7. Apply each firewall filter to an interface to evaluate incoming traffic:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# set family (inet | inet6) filter input firewall-filter-name
```

NOTE: Active flow monitoring is supported only on incoming traffic. You cannot apply firewall filters to evaluate outgoing traffic.

8. Configure the remote server, where GRE encapsulated packets are sent, to perform flow processing.

RELATED DOCUMENTATION

Configuring Port Mirroring

hosted-services

port-mirroring

server-profile (Active Flow Monitoring)

Firewall Filter Nonterminating Actions

Configuring Actively Monitored Interfaces on MX Series Routers

Configure the input interfaces and apply the firewall filter that you defined earlier. Unlike passive flow monitoring, the input interfaces for active flow monitoring are not restricted, so you can select most standard network interfaces (such as ATM1 or Ethernet-based interfaces) as the input.

If you configure active flow monitoring with sampling, you can configure an interface filter in place of a firewall filter with the `sampling` statement at the `[edit interfaces` *`interface-name-fpc/pic/port`* `unit` *`unit-number`* `family inet]` hierarchy level.

```
[edit]
interfaces {
    ge-2/2/0 {
        unit 0 {
            family inet {
                filter {
                    input active_filter;
                }
                address 10.36.11.2/32 {
                    destination 10.36.11.1;
                }
                sampling {
                    (input | output | [input output]);
                }
            }
        }
    }
}
```

Collecting Flow Records

Traffic flows can be exported in flow monitoring version 5, 8, and 9 formats for active flow monitoring. The default export format for flow monitoring records is version 5. To change the export format to flow monitoring version 8, include the `version 8` statement at either the [**edit forwarding-options accounting** ***name*** **output flow-server** ***flow-server-address***] or the [**edit forwarding-options sampling output flow-server** ***flow-server-address***] hierarchy level. To change the export format to flow monitoring version 9, include the `version9 template` *`template-name`* statement at the [**edit forwarding-options sampling output flow-server** ***flow-server-address***] hierarchy level. For more information on flow record formats, see "Flow Monitoring Output Formats" on page 11.

To capture flow data generated by the Monitoring Services PIC, Adaptive Services PIC, or MultiServices PIC and export it to a flow server, you can use one of the following active flow monitoring methods:

- "Configuring M, MX and T Series Routers for Discard Accounting with a Sampling Group" on page 91
- "Configuring M, MX and T Series Routers for Discard Accounting with an Accounting Group" on page 90
- "Configuring M, MX and T Series Routers for Discard Accounting with a Template" on page 93
- "Replicating M, MX and T Series Routing Engine-Based Sampling to Multiple Flow Servers" on page 97
- "Replicating Version 9 Flow Aggregation From M, MX and T Series Routers to Multiple Flow Servers" on page 98
- "Configuring Routing Engine-Based Sampling on M, MX and T Series Routers for Export to Multiple Flow Servers" on page 99
- "Configuring an Aggregate Export Timer on M, MX and T Series Routers for Version 8 Records" on page 127

Configuring M, MX and T Series Routers for Discard Accounting with an Accounting Group

To perform discard accounting on specified traffic, you can collect flow records with the `accounting` statement at the `[edit forwarding-options]` hierarchy level. Like sampling, your topology must be simple (for example, one input interface and one export interface).

Again, you can collect flow records by specifying input and output interfaces. You can configure the input interface to perform discard accounting by applying a firewall filter that contains the `then discard accounting` statement. This match condition directs the filtered traffic to be converted into flow records and exported for analysis by the monitoring services or adaptive services interface. The original packets are then sent to the discard process. For the output, remember to specify the IP address and port of your flow server and the services interface you plan to use for processing flow records.

You must configure a source address, but the **engine-id** and **engine-type** output interface statements are added automatically. You can override these values manually to track different flows with a single flow collector. SNMP input and output interface index information is captured in flow records by default when you configure discard accounting.

```
[edit]
forwarding-options {
    accounting counter1 {
        output {
            flow-inactive-timeout 65;
            flow-active-timeout 65;
            flow-server 10.60.2.1 {
                port 2055;
                version 8;
                aggregation {
                    protocol-port;
                    source-destination-prefix;
                }
            }
            interface sp-2/0/0 {
                engine-id 1;
                engine-type 11;
                source-address 10.60.2.2;
            }
        }
    }
}
```

Configuring M, MX and T Series Routers for Discard Accounting with a Sampling Group

If your needs for active flow monitoring are simple, you can collect flow records with a sampling group. Sampling does not require you to configure a monitoring group (as required in passive flow monitoring) because you can configure flow server information in the **sampling** hierarchy. When you wish to sample traffic, include the `sampling` statement at the `[edit forwarding-options]` hierarchy level.

The typical sampling configuration has one input interface and one export interface. The input interface is activated by the `then sample` statement in a firewall filter term. This match condition directs traffic to the sampling process. Alternatively, you can use an interface-based filter in place of a firewall filter if you include the `sampling` statement at the `[edit interfaces` *`interface-name-fpc/pic/port`* `unit` *`unit-number`* `family inet]` hierarchy level.

There are two types of sampling available: PIC-based sampling and Routing Engine-based sampling. PIC-based sampling occurs when a monitoring services or adaptive services interface is the target for the output of the sampling process. To enable PIC-based sampling, include the `interface` statement at the `[edit forwarding-options sampling output]` hierarchy level and specify a monitoring services or adaptive services interface as the output interface. If an output interface is not specified in the sampling configuration, sampling is performed by the Routing Engine.

To specify a flow server in a sampling configuration, include the `flow-server` statement at the `[edit forwarding-options sampling output]` hierarchy level. You must specify the IP address, port number, and flow monitoring version of the destination flow server. Routing Engine-based sampling supports flow aggregation of up to eight flow servers (version 5 servers and version 8 only) at a time. The export packets are replicated to all flow servers configured to receive them. In contrast, PIC-based sampling allows you to specify just one version 5 flow server and one version 8 server simultaneously. Flow servers operating simultaneously must have different IP addresses.

As part of the output interface statements, you must configure a source address. In contrast, the interface-level statements of **engine-id** and **engine-type** are both added automatically. However, you can override these values with manually configured statements to track different flows with a single flow collector, as needed. When you configure sampling, SNMP input and output interface index information is captured in flow records by default.

```
[edit]
forwarding-options {
    sampling {
        input {
            rate 1;
        }
        family inet {
            output {
                flow-inactive-timeout 15;
                flow-server 10.60.2.1 {
                    port 2055;
                    version 5;
                }
                interface sp-2/0/0 {
                    engine-id 5;
                    engine-type 55;
                    source-address 10.60.2.2;
                }
            }
        }
```

```
    }
}
```

Configuring M, MX and T Series Routers for Discard Accounting with a Template

Flow monitoring version 9, which is based on RFC 3954, provides a way to organize flow data into templates. Version 9 also provides a way to actively monitor IPv4, IPv6, MPLS, and peer AS billing traffic. Version 9 is not supported on the AS-I PIC.

To activate templates in flow monitoring, you must configure a template and include that template in the version 9 flow monitoring configuration. Version 9 does not work in conjunction with versions 5 and 8.

To configure a version 9 template, include the **template** *template-name* statement at the [**edit services flow-monitoring version9**] hierarchy level. The Junos OS supports five different templates: **ipv4-template**, **ipv6–template**, **mpls-template**, **mpls-ipv4-template**, and **peer-as-billing-template**. To view the fields selected in each of these templates, see "Flow Monitoring Version 9 Format Output Fields" on page 26.

```
[edit services]
flow-monitoring {
    version9 { # Specifies flow monitoring version 9.
        template mpls { # Specifies template you are configuring.
        template-refresh-rate {
            packets 6000; # The default is 4800 packets and the range is 1-480000 packets.
            seconds 90; # The default is 60 seconds and the range is 1-600 seconds.
         }
         option--refresh-rate {
             packets 3000; # The default is 4800 packets and the range is 1-480000 # packets.
             seconds 30; # The default is 60 seconds and the range is 1-600.
         }
         flow-active-timeout 60;  # The default is 60 seconds and the range is # 10-600.
         flow-inactive-timeout 30; # The default is 60 seconds and the range 10-600.
         template-refresh-rate seconds 10; # The default is 600 seconds and the # range is 10-600
         mpls-template {
             label-positions [1 | 2 | 3]; # Specifies label position for the MPLS template.
         }
     }
}
```

You can export to multiple templates at a time to a maximum of eight flow servers for AS PICs and one flow server for all other PICs. To assign a template to a flow output, include the `template` *`template-name`* statement at the `[edit forwarding options sampling output flow-server version9]` hierarchy level:

```
[edit]
forwarding-options {
    sampling {
        input {
            family mpls {
                rate 1;
                run-length 1;
            }
        }
        output {
            flow-server 10.60.2.1 { # The IP address and port of the flow server.
                port 2055;
                source-address 192.0.2.1;
                version9 { # Records are sent to the flow server using version 9 format.
                    template { # Indicates a template will organize records.
                        mpls; # Records are sent to the MPLS template.
                    }
                }
            }
        }
    }
}
```

Defining a Firewall Filter on MX Series Routers to Select Traffic for Active Flow Monitoring

The first step in active flow monitoring is to configure the match conditions for acceptable traffic or quarantined traffic. Common match actions for active flow monitoring include `sample`, `discard accounting`, `port-mirror`, and `accept`. To configure, include the desired action statements and a counter as part of the `then` statement in a firewall filter and apply the filter to an interface.

In sampling, the router reviews a portion of the traffic and sends reports about this sample to the flow monitoring server. Discard accounting traffic is counted and monitored, but not forwarded out of the router. Port-mirrored traffic is copied and sent to another interface. Accepted traffic is forwarded to the intended destination.

Most of these match combinations are valid. However, you can either port-mirror or sample with the same traffic at the same time, but not perform more than one action simultaneously on the same packets.

```
[edit]
firewall {
    family inet {
        filter active_filter {
            term quarantined_traffic {
                from {
                    source-address {
                        10.36.1.2/32;
                    }
                }
                then {
                    count quarantined-counter;
                    sample;
                    discard accounting;
                }
            }
            term copy_and_forward_the_rest {
                then {
                    port-mirror;
                    accept;
                }
            }
        }
    }
}
```

Processing IPv4 traffic on an MX Series Router Using Monitoring Services, Adaptive Services or Multiservices Interfaces

You configure the monitoring services, adaptive services, or multiservices interfaces with the `family inet` statement so they can process IPv4 traffic. However, you must remember that a monitoring services interface uses an **mo-** prefix and adaptive services and multiservices interfaces use an **sp-** prefix.

```
[edit]
interfaces {
    sp-2/0/0 {
        unit 0 {
            family inet {
                address 10.36.100.1/32 {
                    destination 10.36.100.2;
                }
            }
        }
    }
}
```

Active flow monitoring records leave the router through an export interface to reach the flow monitoring server.

```
[edit]
interfaces {
    fe-1/0/0 {
        unit 0 {
            family inet {
                address 10.60.2.2/30;
            }
        }
    }
}
```

Replicating M, MX and T Series Routing Engine-Based Sampling to Multiple Flow Servers

Routing Engine-based sampling supports up to eight flow servers for both flow monitoring version 5 and version 8 configurations. The total number of flow servers is limited to eight, regardless of how many are configured for version 5 or version 8.

When you configure version 5 or version 8 sampling, the export packets are replicated to all flow servers configured to receive them. If two flow servers are configured to receive version 5 records, both flow servers will receive records for a specified flow.

NOTE: With Routing-Engine-based sampling, if multiple flow servers are configured with version 8 export format, all of them must use the same aggregation type (for example, all flow servers receiving version 8 export could be configured for source-destination aggregation type).

The following configuration example allows replication of export packets to two flow servers.

```
[edit]
forwarding-options {
    sampling {
        input {
            rate 1;
            }
        }
        output {
            flow-server 10.10.3.2 {
                port 2055;
                version 5;
                source-address 192.168.164.119;
            }
            flow-server 172.17.20.62 {
                port 2055;
                version 5;
                source-address 192.168.164.119;
            }
        }
    }
}
```

Replicating Version 9 Flow Aggregation From M, MX and T Series Routers to Multiple Flow Servers

With this feature, you can configure up to eight flow servers to receive packets for a version 9 flow monitoring template. Once a flow server is configured to receive this data, it will also receive the following periodic version 9 flow monitoring updates:

- Options data
- Template definition

With Routing Engine-based sampling, if multiple collectors are configured with version 8 export format, all of them must use the same aggregation-type.

The option and template definition refresh period is configured on a per-template basis at the [**edit services flow-monitoring**] hierarchy level.

The following configuration example allows replication of version 9 export packets to two flow servers.

```
forwarding-options {
    sampling {
        input {
            family inet {
                rate 1;
            }
        }
        output {
            flow-server 10.10.3.2 {
                port 2055;
                version9 {
                    template {
                        ipv4;
                    }
                }
            }
            flow-server 172.17.20.62 {
                port 2055;
                version9 {
                    template {
                        ipv4;
                    }
                }
            }
```

```
            flow-inactive-timeout 30;
            flow-active-timeout 60;
            interface sp-4/0/0 {
                source-address 10.10.3.4;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Active Flow Monitoring Overview | 54

Active Flow Monitoring Overview | 53

Active Flow Monitoring Applications | 46

Replicating M, MX and T Series Routing Engine-Based Sampling to Multiple Flow Servers | 97

Configuring Routing Engine-Based Sampling on M, MX and T Series Routers for Export to Multiple Flow Servers

Routing Engine-based sampling supports up to eight flow servers for both version 5 and version 8 configurations. The total number of collectors is limited to eight, regardless of how many are configured for version 5 or version 8. When you configure sampling, the export packets are replicated to all collectors configured to receive them. If two collectors are configured to receive version 5 records, both collectors will receive records for a specified flow.

The following configuration example allows replication of export packets to two collectors.

```
forwarding-options {
    sampling {
        input {
            family inet {
                rate 1;
            }
        }
        output {
            cflowd 10.10.3.2 {
                port 2055;
```

```
                version 5;
                source-address 192.168.164.119;
            }
            cflowd 172.17.20.62 {
                port 2055;
                version 5;
                source-address 192.168.164.119;
            }
        }
    }
}
```

Example: Copying Traffic to a PIC While an M, MX or T Series Router Forwards the Packet to the Original Destination

IN THIS SECTION

Traffic sampling enables you to copy traffic to a Physical Interface Card (PIC) while the router forwards the packet to its original destination. This example describes how to configure a router to perform sampling on the Routing Engine using the **sampled** process. For this method, you configure a filter (input or output) with a matching term that contains the `then sample` statement. In addition, for VPN routing and forwarding (VRF) Routing Engine-based sampling, you configure a VRF routing instance that maps to an interface. Each VRF instance corresponds with a forwarding table. Routes on the interface go into the corresponding forwarding table.

For VRF Routing Engine-based sampling, the kernel queries the correct VRF route table based on the ingress interface index for the received packet. For interfaces configured in VRF, the sampled packets contain the correct input and output interface SNMP index, the source and destination AS numbers, and the source and destination mask.

NOTE: With Junos OS Release 10.1, VRF Routing Engine-based sampling is performed only on IPv4 traffic. You cannot use Routing Engine-based sampling on IPv6 traffic or on MPLS label-switched paths.

This example describes how to configure and verify VRF Routing Engine-based sampling on one router in a four-router topology.

Requirements

This example uses the following hardware and software components:

- Junos OS Release 10.1 or later
- M Series, MX Series, or T Series router

Before you configure VRF Routing Engine-Based sampling on your router, be sure you have an active connection between the routers on which you configure sampling. In addition, you need to have an understanding of VRF to configure the interfaces and routing instances that form the basis of the sampling configuration; and an understanding of the BGP, MPLS, and OSPF protocols to configure the other routers in the network to bring up the sampling configuration.

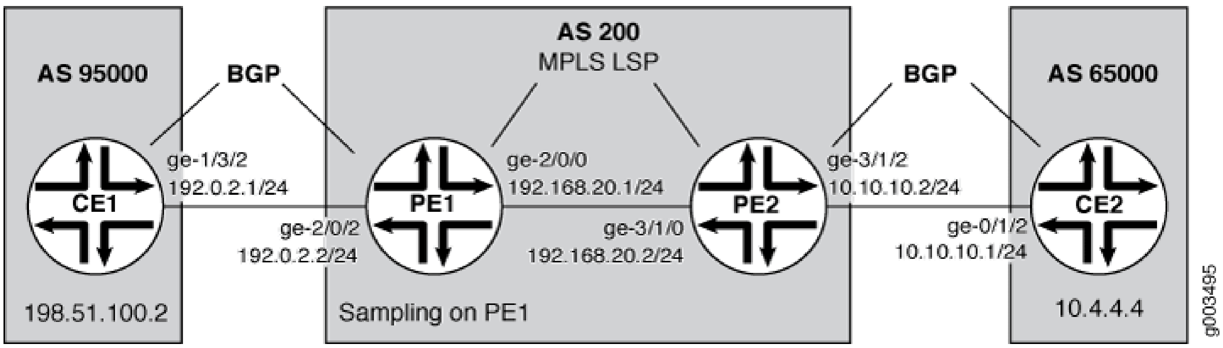
Overview and Topology

IN THIS SECTION

The scenario in this example illustrates VRF Routing Engine-based sampling configured on the PE1 router in a four-router network. The CE routers use BGP as the routing protocol to communicate with the PE routers. MPLS LSPs pass traffic between the PE routers. Packets from the CE1 router are sampled on the PE1 router. Regular traffic is forwarded to the original destination (the CE2 router).

Topology

Figure 16: Routing Engine-Based Sampling Network Topology



Configuration

IN THIS SECTION

In this configuration example, the VRF Routing Engine-based sampling is configured on the PE1 router that samples the traffic that goes through the interface and routes configured in the VRF. The configurations on the other three routers are included to show the sampling configuration on the PE1 router working in the context of a network.

To configure VRF Routing Engine-based sampling for the network example, perform these tasks:

Configuring the CE1 Router

Step-by-Step Procedure

In this step, you configure interfaces, routing options, protocols, and policy options for the CE1 router. To configure the CE1 router:

1. Configure one interface with two IP addresses. One address is for traffic to the PE1 router; the other address is to check that traffic is flowing to the CE2 router:

```
[edit interfaces]
user@router-ce1# set ge-1/3/2 unit 0 family inet address 192.0.2.1/24
user@router-ce1# set ge-1/3/2 unit 0 family inet address 198.51.100.2/8
```

2. Configure the autonomous system to establish a connection between BGP peers:

```
[edit routing-options]
user@router-ce1# set autonomous-system 95000
```

3. Configure BGP as the routing protocol between the CE router and the PE router:

```
[edit protocols]
user@router-ce1# set bgp group to_r1 type external
user@router-ce1# set bgp group to_r1 export my_lo0_addr
user@router-ce1# set bgp group to_r1 peer-as 200
user@router-ce1# set bgp group to_r1 neighbor 192.0.2.2
```

4. Configure the policies that ensure that the CE routers exchange routing information. In this example, Router CE1 exchanges routing information with Router CE2:

```
[edit policy-options]
user@router-ce1# set policy-statement my_lo0_addr term one from protocol direct
user@router-ce1# set policy-statement my_lo0_addr term one from route-filter 10.255.15.32/32
exact
user@router-ce1# set policy-statement my_lo0_addr term one then accept
user@router-ce1# set policy-statement my_lo0_addr term four from protocol direct
user@router-ce1# set policy-statement my_lo0_addr term four from route-filter 203.0.113.0/8
exact
user@router-ce1# set policy-statement my_lo0_addr term four then accept
```

Results

The output below shows the configuration of the CE1 router:

```
[edit]
user@router-ce1# show
[...Output Truncated...]
interfaces {
    ge-1/3/2 {
        unit 0 {
            family inet {
                address 192.0.2.1/24;
                address 198.51.100.2/8;
            }
        }
    }
}
routing-options {
    autonomous-system 95000;
}
protocols {
    bgp {
        group to_r1 {
            type external;
            export my_lo0_addr;
            peer-as 200;
            neighbor 192.0.2.2;
        }
    }
}
policy-options {
    policy-statement my_lo0_addr {
        term one {
            from {
                protocol direct;
                route-filter 10.255.15.32/32 exact;
            }
            then accept;
        }
        term four {
            from {
                protocol direct;
```

```
                route-filter 203.0.113.0/8 exact;
            }
            then accept;
        }
    }
}
```

Configuring the PE1 Router

Step-by-Step Procedure

In this step, you configure a filter with a matching term that contains the `then sample` statement and apply the filter to the ingress interface. You also configure a VRF routing instance with import and export policies. In addition, you configure interfaces, forwarding options, routing options, protocols, and policy options for the PE1 router. To configure the PE1 router:

1. Create the **fw** firewall filter that is applied to the logical interface being sampled:

```
[edit firewall]
user@router-pe1# set family inet filter fw term 1 from protocol tcp
user@router-pe1# set family inet filter fw term 1 from port bgp
user@router-pe1# set family inet filter fw term 1 then accept
user@router-pe1# set family inet filter fw term 2 then sample
```

2. Configure two interfaces, one interface that connects to the CE1 router (**ge-2/0/2**), and another that connects to the PE2 router (**ge-2/0/0**):

```
[edit interfaces]
user@router-pe1# set ge-2/0/2 unit 0 family inet address 192.0.2.2/24
user@router-pe1# set ge-2/0/0 unit 0 family inet address 192.168.20.1/24
user@router-pe1# set ge-2/0/0 unit 0 family mpls
```

3. Enable MPLS on the interface that connects to the PE2 router (**ge-2/0/0**):

```
[edit interfaces]
user@router-pe1# set ge-2/0/0 unit 0 family mpls
```

4. On the interface that connects to the CE1 router, apply the **fw** filter that was configured in the firewall configuration:

```
[edit interfaces]
user@router-pe1# set ge-2/0/2 unit 0 family inet filter input fw
user@router-pe1# set ge-2/0/2 unit 0 family inet filter output fw
```

5. Configure the management (**fxp0**) and loopback (**lo0**) interfaces:

```
[edit interfaces]
user@router-pe1# set fxp0 unit 0 family inet address 192.168.69.153/21
user@router-pe1# set lo0 unit 0 family inet address 127.0.0.1/32
```

6. Configure the **sampled** log file in the **/var/log** directory to record traffic sampling:

```
[edit forwarding-options]
user@router-pe1# set sampling traceoptions file sampled
user@router-pe1# set sampling traceoptions file world-readable
user@router-pe1# set sampling traceoptions flag all
```

7. Specify the sampling rate and threshold value for traffic sampling:

```
[edit forwarding-options]
user@router-pe1# set sampling input rate 1
user@router-pe1# set sampling input run-length 0
user@router-pe1# set sampling input max-packets-per-second 20000
```

8. Specify active and inactive flow periods, and the router (**198.51.100.2**) that sends out the monitored information:

```
[edit forwarding-options]
user@router-pe1# set sampling family inet output flow-active-timeout 60
user@router-pe1# set sampling family inet output flow-inactive-timeout 60
user@router-pe1# set sampling family inet output flow-server 198.51.100.2 port 2055
user@router-pe1# set sampling family inet output flow-server 198.51.100.2 local-dump
user@router-pe1# set sampling family inet output flow-server 198.51.100.2 version 500
```

9. Configure the autonomous system to establish a connection between BGP peers:

```
[edit routing-options]
user@router-pe1# set autonomous-system 200
```

10. Configure RSVP to support MPLS label-switched paths (LSPs) between the PE routers:

```
[edit protocols]
user@router-pe1# set rsvp interface all
user@router-pe1# set rsvp interface fxp0.0 disable
```

11. Configure an MPLS LSP from the PE1 router to the PE2 router:

```
[edit protocols]
user@router-pe1# set mpls label-switched-path R1toR2 from 192.168.20.1
user@router-pe1# set mpls label-switched-path R1toR2 to 192.168.20.2
user@router-pe1# set mpls interface all
user@router-pe1# set mpls interface fxp0.0 disable
```

12. Configure an internal BGP group for the PE routers. Include the `family inet-vpn unicast` statement to enable BGP to carry network layer reachability information (NLRI) parameters and for BGP peers to only carry unicast routes for forwarding:

```
[edit protocols]
user@router-pe1# set bgp group to_r2 type internal
user@router-pe1# set bgp group to_r2 local-address 192.168.20.1
user@router-pe1# set bgp group to_r2 neighbor 192.168.20.2 family inet-vpn unicast
```

13. Configure OSPF as the interior gateway protocol (IGP) and to compute the MPLS LSPs:

```
user@router-pe1# set ospf traffic-engineering
user@router-pe1# set ospf area 0.0.0.0 interface all
user@router-pe1# set ospf area 0.0.0.0 interface fxp0.0 disable
```

14. Create the extended community that is applied in the policy options configuration:

```
[edit policy-options]
user@router-pe1# set community vpna-comm members target:200:100
```

15. Define the **vpna-export** routing policy that is applied in the `vrf-export` statement in the routing instance configuration. Also, apply the **vpna-comm** community from which routes are learned:

```
[edit policy-options]
user@router-pe1# set policy-statement vpna-export term one from protocol bgp
user@router-pe1# set policy-statement vpna-export term one from protocol direct
user@router-pe1# set policy-statement vpna-export term one then community add vpna-comm
user@router-pe1# set policy-statement vpna-export term one then accept
user@router-pe1# set policy-statement vpna-export term two then reject
```

16. Define the **vpna-import** routing policy that is applied in the `vrf-import` statement in the routing instance configuration. Also, apply the **vpna-comm** community from which routes are learned:

```
[edit policy-options]
user@router-pe1# set policy-statement vpna-import term one from protocol bgp
user@router-pe1# set policy-statement vpna-import term one from community vpna-comm
user@router-pe1# set policy-statement vpna-import term one then accept
user@router-pe1# set policy-statement vpna-import term two then reject
```

17. Configure a VRF routing instance so that routes received from the provider edge-provider edge (PE-PE) session can be imported into any of the instance's VRF secondary routing tables:

```
 [edit routing-instances]
user@router-pe1#set vrf1 instance-type vrf set vrf1 interface ge-2/0/2.0
user@router-pe1# set vrf1 route-distinguisher 10.255.15.51:1
user@router-pe1# set vrf1 vrf-import vpna-import
user@router-pe1# set vrf1 vrf-export vpna-export
user@router-pe1# set vrf1 protocols bgp group customer type external
user@router-pe1# set vrf1 protocols bgp group customer peer-as 95000
user@router-pe1# set vrf1 protocols bgp group customer as-override
user@router-pe1# set vrf1 protocols bgp group customer neighbor 192.168.30.1
user@router-pe1# set vrf1 protocols bgp group customer neighbor 192.0.2.1
```

Results

Check the results of the configuration for the PE1 router:

```
user@router-pe1> show configuration
[...Output Truncated...]
}
interfaces {
    ge-2/0/0 {
        unit 0 {
            family inet {
                address 192.168.20.1/24;
            }
            family mpls;
        }
    }
    ge-2/0/2 {
        unit 0 {
            family inet {
                filter {
                    input fw;
                    output fw;
                }
                address 192.0.2.2/24;
            }
        }
    }
    fxp0 {
        unit 0 {
            family inet {
                address 192.168.69.153/21;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 127.0.0.1/32;
            }
        }
    }
}
```

```
forwarding-options {
    sampling {
        traceoptions {
            file sampled world-readable;
            flag all;
        }
        input {
            rate 1;
            run-length 0;
            max-packets-per-second 20000;
        }
        family inet {
            output {
                flow-inactive-timeout 60;
                flow-active-timeout 60;
                flow-server 198.51.100.2 {
                    port 2055;
                    local-dump;
                    version 500;
                }
            }
        }
    }
}
routing-options {
[...Output Truncated...]
    autonomous-system 200;
}
protocols {
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        label-switched-path R1toR2 {
            from 192.168.20.1;
            to 192.168.20.2;
        }
        interface all;
        interface fxp0.0 {
            disable;
```

```
        }
    }
    bgp {
        group to_r2 {
            type internal;
            local-address 192.168.20.1;
            neighbor 192.168.20.2 {
                family inet-vpn {
                    unicast;
                }
            }
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
}
policy-options {
    policy-statement vpna-export {
        term one {
            from protocol [ bgp direct ];
            then {
                community add vpna-comm;
                accept;
            }
        }
        term two {
            then reject;
        }
    }
    policy-statement vpna-import {
        term one {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
```

```
        }
        term two {
            then reject;
        }
    }
    community vpna-comm members target:200:100;
}
firewall {
    family inet {
        filter fw {
            term 1 {
                from {
                    protocol tcp;
                    port bgp;
                }
                then accept;
            }
            term 2 {
                then sample;
            }
        }
    }
}
routing-instances {
    vrf1 {
        instance-type vrf;
        interface ge-2/0/2.0;
        route-distinguisher 10.255.15.51:1;
        vrf-import vpna-import;
        vrf-export vpna-export;
        protocols {
            bgp {
                group customer {
                    type external;
                    peer-as 95000;
                    as-override;
                    neighbor 192.168.30.1;
                    neighbor 192.0.2.1;
                }
            }
        }
```

```
    }
}
```

Configuring the PE2 Router

Step-by-Step Procedure

In this step, you configure a filter with a matching term that contains the `then sample` statement and apply the filter to the ingress interface. You also configure a VRF routing instance with import and export policies. In addition, you configure interfaces, forwarding options, routing options, protocols, and policy options for the PE2 router. To configure the PE2 router:

1. Create the **fw** firewall filter that is applied to the logical interface being sampled:

   ```
   [edit firewall]
   user@router-pe2# set family inet filter fw term 1 from protocol tcp
   user@router-pe2# set family inet filter fw term 1 from port bgp
   user@router-pe2# set family inet filter fw term 1 then accept
   user@router-pe2# set family inet filter fw term 2 then sample
   user@router-pe2# set family inet filter fw term 2 then accept
   ```

2. Configure two interfaces, one interface that connects to the CE2 router (**ge-3/1/2**), and another that connects to the PE1 router (**ge-3/1/0**):

   ```
   [edit interfaces]
   user@router-pe2# set ge-3/1/0 unit 0 family inet address 192.168.20.2/24
   user@router-pe2#  set ge-3/1/0 unit 0 family mpls
   user@router-pe2#  set ge-3/1/2 unit 0 family inet address 10.10.10.2/24
   ```

3. Enable MPLS on the interface that connects to the PE1 router (**ge-3/1/0**):

   ```
   [edit interfaces]
   user@router-pe2#  set ge-3/1/0 unit 0 family mpls
   ```

4. On the interface that connects to the CE2 router, apply the **fw** filter that was configured in the firewall configuration:

```
[edit interfaces]
user@router-pe2#  set ge-3/1/2 unit 0 family inet filter input fw
user@router-pe2#  set ge-3/1/2 unit 0 family inet filter output fw
```

5. Configure the **sampled** log file in the **/var/log** directory to record traffic sampling:

```
[edit forwarding-options]
user@router-pe2# set sampling traceoptions file sampled
user@router-pe2# set sampling traceoptions file world-readable
user@router-pe1# set sampling traceoptions flag all
```

6. Specify the sampling rate and threshold value for traffic sampling:

```
[edit forwarding-options]
user@router-pe2# set sampling input rate 1
user@router-pe2# set sampling input run-length 0
user@router-pe2# set sampling input max-packets-per-second 20000
```

7. Specify active and inactive flow periods, and the router (**198.51.100.2**) that sends out the monitored information:

```
[edit forwarding-options]
user@router-pe2# set sampling family inet output flow-active-timeout 60
user@router-pe2# set sampling family inet output flow-inactive-timeout 60
user@router-pe2# set sampling family inet output flow-server 198.51.100.2 port 2055
user@router-pe2# set sampling family inet output flow-server 198.51.100.2 local-dump
user@router-pe2# set sampling family inet output flow-server 198.51.100.2 version 500
```

8. Configure the autonomous system to establish a connection between BGP peers:

```
[edit routing-options]
user@router-pe2# set autonomous-system 200
```

9. Configure RSVP to support MPLS label-switched paths (LSPs) between the PE routers:

```
[edit protocols]
user@router-pe2# set rsvp interface all
user@router-pe2# set rsvp interface fxp0.0 disable
```

10. Configure an MPLS LSP from the PE2 router to the PE1 router:

```
[edit protocols]
user@router-pe2# set mpls label-switched-path R2toR1 from 192.168.20.2
user@router-pe2# set mpls label-switched-path R2toR1 to 192.168.20.1
user@router-pe2# set mpls interface all
user@router-pe2# set mpls interface fxp0.0 disable
```

11. Configure an internal BGP group for the PE routers. Include the `family inet-vpn unicast` statement to enable BGP to carry network layer reachability information (NLRI) parameters and for BGP peers to only carry unicast routes for forwarding:

```
[edit protocols]
user@router-pe2# set bgp group to_r1 type internal
user@router-pe2# set bgp group to_r1 local-address 192.168.20.2
user@router-pe2# set bgp group to_r1 neighbor 192.168.20.1 family inet-vpn unicast
```

12. Configure OSPF as the interior gateway protocol (IGP) and to compute the MPLS LSPs:

```
[edit protocols]
user@router-pe2# set ospf traffic-engineering
user@router-pe2# set ospf area 0.0.0.0 interface all
user@router-pe2# set ospf area 0.0.0.0 interface fxp0.0 disable
```

13. Create the extended community that is applied in the policy options configuration:

```
[edit policy-options]
user@router-pe2# set community vpna-comm members target:200:100
```

14. Define the **vpna-export** routing policy that is applied in the `vrf-export` statement in the routing instance configuration. Also, apply the **vpna-comm** community from which routes are learned:

```
[edit policy-options]
user@router-pe2# set policy-statement vpna-export term one from protocol bgp
user@router-pe2# set policy-statement vpna-export term one from protocol direct
user@router-pe2# set policy-statement vpna-export term one then community add vpna-comm
user@router-pe2# set policy-statement vpna-export term one then accept
user@router-pe2# set policy-statement vpna-export term two then reject
```

15. Define the **vpna-import** routing policy that is applied in the `vrf-import` statement in the routing instance configuration. Also, apply the **vpna-comm** community from which routes are learned:

```
[edit policy-options]
user@router-pe2# set policy-statement vpna-import term one from protocol bgp
user@router-pe2# set policy-statement vpna-import term one from community vpna-comm
user@router-pe2# set policy-statement vpna-import term one then accept
user@router-pe2# set policy-statement vpna-import term two then reject
```

16. Configure a VRF routing instance so that routes received from the provider edge-provider edge (PE-PE) session can be imported into any of the instance's VRF secondary routing tables:

```
 [edit routing-instances]
user@router-pe2# set vrf1 instance-type vrf
 user@router-pe2# set vrf1 interface ge-3/1/2.0
 user@router-pe2# set vrf1 route-distinguisher 10.255.19.12:1
user@router-pe2# set vrf1 vrf-import vpna-import
 user@router-pe2# set vrf1 vrf-export vpna-export
user@router-pe2#set vrf1 protocols bgp group R3-R4 type external
 user@router-pe2# set vrf1 protocols bgp group R3-R4 peer-as 65000
 user@router-pe2# set vrf1 protocols bgp group R3-R4 as-override
user@router-pe2# set vrf1 protocols bgp group R3-R4 neighbor 10.10.10.1
```

Results

Check the results of the configuration for the PE2 router:

```
user@router-pe2> show configuration
[...Output Truncated...]
}
interfaces {
    ge-3/1/0 {
        unit 0 {
            family inet {
                address 192.168.20.2/24;
            }
            family mpls;
        }
    }
    ge-3/1/2 {
        unit 0 {
            family inet {
                filter {
                    input fw;
                    output fw;
                }
                address 10.10.10.2/24;
            }
        }
    }
}
forwarding-options {
    sampling {
        traceoptions {
            file sampled world-readable;
            flag all;
        }
        input {
            rate 1;
            run-length 0;
            max-packets-per-second 20000;
        }
        family inet {
            output {
                flow-inactive-timeout 60;
```

```
                flow-active-timeout 60;
                flow-server 198.51.100.2 {
                    port 2055;
                    local-dump;
                    version 500;
                }
            }
        }
    }
}
routing-options {
[...Output Truncated...]
    autonomous-system 200;
}
protocols {
    rsvp {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    mpls {
        label-switched-path R2toR1 {
            from 192.168.20.2;
            to 192.168.20.1;
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
    bgp {
        group to_r1 {
            type internal;
            local-address 192.168.20.2;
            neighbor 192.168.20.1 {
                family inet-vpn {
                    unicast;
                }
            }
            neighbor 192.0.2.1;
        }
    }
```

```
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
}
policy-options {
    policy-statement vpna-export {
        term one {
            from protocol [ bgp direct ];
            then {
                community add vpna-comm;
                accept;
            }
        }
        term two {
            then reject;
        }
    }
    policy-statement vpna-import {
        term one {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term two {
            then reject;
        }
    }
    community vpna-comm members target:200:100;
}
firewall {
    family inet {
        filter fw {
            term 1 {
                from {
                    protocol tcp;
```

```
                    port bgp;
                }
                then accept;
            }
            term 2 {
                then {
                    sample;
                    accept;
                }
            }
        }
    }
}
routing-instances {
    vrf1 {
        instance-type vrf;
        interface ge-3/1/2.0;
        route-distinguisher 10.255.19.12:1;
        vrf-import vpna-import;
        vrf-export vpna-export;
        protocols {
            bgp {
                group R3-R4 {
                    type external;
                    peer-as 65000;
                    as-override;
                    neighbor 10.10.10.1;
                }
            }
        }
    }
}
```

Configuring the CE2 Router

Step-by-Step Procedure

In this step, you configure interfaces, routing options, protocols, and policy options for the CE2 router. To configure the CE2 router:

1. Configure one interface with two IP addresses. One address is for traffic to the PE2 router and the other address is to check that traffic is flowing from the CE1 router:

```
[edit interfaces]
user@router-ce2# set ge-0/1/2 unit 0 family inet address 10.10.10.1/24
user@router-ce2#  set ge-0/1/2 unit 0 family inet address 10.4.4.4/16
```

2. Configure the autonomous system to establish a connection between BGP peers:

```
[edit routing-options]
user@router-ce1# set autonomous-system 65000
```

3. Configure BGP as the routing protocol between the CE and the PE routers:

```
[edit protocols]
user@router-ce2# set bgp group R3-R4 type external
 user@router-ce2#  set bgp group R3-R4 export l3vpn-policy
user@router-ce2#  set bgp group R3-R4 peer-as 200
 user@router-ce2#  set bgp group R3-R4 neighbor 10.10.10.2
```

4. Configure the policies that ensure that the CE routers exchange routing information. In this example, Router CE2 exchanges routing information with Router CE1:

```
[edit policy-options]
user@router-ce2# set policy-statement l3vpn-policy term one from protocol direct
 user@router-ce2#  set policy-statement l3vpn-policy term one from route-filter
10.255.15.75/32 exact
user@router-ce2# set policy-statement l3vpn-policy term one then accept
user@router-ce2# set policy-statement l3vpn-policy term two from protocol direct
user@router-ce2# set policy-statement l3vpn-policy term two from route-filter 10.4.0.0/16
exact
 user@router-ce2#  set policy-statement l3vpn-policy term two then accept
```

Results

The output below shows the configuration of the CE2 router:

```
[edit]
user@router-ce2# show
[...Output Truncated...]
interfaces {
    ge-0/1/2 {
        unit 0 {
            family inet {
                address 10.10.10.1/24;
                address 10.4.4.4/16;
            }
        }
    }
}
routing-options {
    autonomous-system 65000;
}
protocols {
    bgp {
        group R3-R4 {
            type external;
            export l3vpn-policy;
            peer-as 200;
            neighbor 10.10.10.2;
        }
    }
}
policy-options {
    policy-statement l3vpn-policy {
        term one {
            from {
                protocol direct;
                route-filter 10.255.15.75/32 exact;
            }
            then accept;
        }
        term two {
            from {
                protocol direct;
```

```
                  route-filter 10.4.0.0/16 exact;
              }
              then accept;
          }
      }
  }
```

Verification

IN THIS SECTION

After you have completed the configuration of the four routers, you can verify that traffic is flowing from the CE1 router to the CE2 router, and you can observe the sampled traffic from two locations. To confirm that the configuration is working properly, perform these tasks:

Verifying the Traffic Flow Between the CE Routers

Purpose

Use the `ping` command to verify traffic between the CE routers.

Action

From the CE1 router, issue the `ping` command to the CE2 router:

```
user@router-ce2> ping 10.4.4.4 source 198.51.100.2
PING 10.4.4.4 (10.4.4.4): 56 data bytes
64 bytes from 10.4.4.4: icmp_seq=0 ttl=64 time=0.861 ms
64 bytes from 10.4.4.4: icmp_seq=1 ttl=64 time=0.869 ms
64 bytes from 10.4.4.4: icmp_seq=2 ttl=64 time=0.786 ms
^C
--- 10.4.4.4 ping statistics ---
```

```
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.786/0.839/0.869/0.037 ms
```

Meaning

The output from the `ping` command shows that the `ping` command was successful. Traffic is flowing between the CE routers.

Verifying Sampled Traffic

Purpose

You can observe the sampled traffic using the `show log sampled` command from the CLI or from the router shell using the `tail -f /var/log/sampled` command. In addition, you can collect the logs in a flowcollector. The same information appears in the output of both commands and in the flow collector. For information about using a flow collector, see “"Sending cflowd Records to Flow Collector Interfaces" on page 234” and “"Example: Configuring a Flow Collector Interface on an M, MX or T Series Router" on page 203.”

Action

From the PE1 router, use the `show log sampled` command:

```
user@router-pe1> show log sampled
[...Output Truncated...]
Nov 16 23:24:19    Src addr: 198.51.100.2
Nov 16 23:24:19    Dst addr: 10.4.4.4
Nov 16 23:24:19    Nhop addr: 192.168.20.2
Nov 16 23:24:19    Input interface: 503     # SNMP index of the incoming interface on PE1
Nov 16 23:24:19    Output interface: 505     # SNMP index of the outgoing interface on PE1
Nov 16 23:24:19    Pkts in flow: 5
Nov 16 23:24:19    Bytes in flow: 420
Nov 16 23:24:19    Start time of flow: 602411369
Nov 16 23:24:19    End time of flow: 602415369
Nov 16 23:24:19    Src port: 0
Nov 16 23:24:19    Dst port: 2048
Nov 16 23:24:19    TCP flags: 0x0
Nov 16 23:24:19    IP proto num: 1
Nov 16 23:24:19    TOS: 0x0
Nov 16 23:24:19    Src AS: 95000     # The autonomous system of CE1
```

```
Nov 16 23:24:19    Dst AS: 65000,,,,,# The autonomous system of CE2
Nov 16 23:24:19    Src netmask len: 8
Nov 16 23:24:19    Dst netmask len: 16
Nov 16 23:24:19 cflowd header:
Nov 16 23:24:19   Num-records: 1
Nov 16 23:24:19   Version: 500
Nov 16 23:24:19   Flow seq num: 13
Nov 16 23:24:19   Sys Uptime: 602450382 (msecs)
Nov 16 23:24:19   Time-since-epoch: 1258413859 (secs)
Nov 16 23:24:19   Engine id: 0
Nov 16 23:24:19   Engine type: 0
Nov 16 23:24:19   Sample interval: 1
[...Output Truncated...]
```

Meaning

The output from the `show log sampled` command shows the correct SNMP index for the incoming and outgoing interfaces on the PE1 router. Also, the source and destination addresses for the autonomous systems for the two CE routers are correct.

Cross Verifying Sampled Traffic

Purpose

You can also double check that the sampled traffic is the correct traffic by using the `show interface` *`interface-name-fpc/pic/port.unit-number`* `| match SNMP` command and the `show route` *`route-name`* `detail` command.

Action

The following output is a cross check of the output in the "Verifying Sampled Traffic" on page 124 task:

```
user@router-pe1> show interfaces ge-2/0/2.0 | match SNMP
  Logical interface ge-2/0/2.0 (Index 76) (SNMP ifIndex 503)
    Flags: SNMP-Traps 0x4000000 Encapsulation: ENET2
```

```
user@router-pe1> show route 10.4.4.4 detail

vrf1.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
10.4.0.0/16 (1 entry, 1 announced)
```

```
         *BGP    Preference: 170/-101
                 Route Distinguisher: 10.255.19.12:1
                 Next hop type: Indirect
                 Next-hop reference count: 6
                 Source: 192.168.20.2
                 Next hop type: Router, Next hop index: 659
                 Next hop: 192.168.20.2 via ge-2/0/0.0 weight 0x1, selected
                 Label operation: Push 299776
                 Protocol next hop: 192.168.20.2
                 Push 299776
                 Indirect next hop: 8e6f780 1048574
                 State: <Secondary Active Int Ext>
                 Local AS:   200 Peer AS:   200
                 Age: 3d 19:49:32     Metric2: 65535
                 Task: BGP_200.20.20.20.2+179
                 Announcement bits (3): 0-RT 1-BGP RT Background 2-KRT
                 AS path: 65000 I
                 AS path: Recorded
                 Communities: target:200:100
                 Import Accepted
                 VPN Label: 299776
                 Localpref: 100
                 Router ID: 10.10.10.2
                 Primary Routing Table bgp.l3vpn.0
```

Meaning

The output of the `show interfaces ge-2/0/2.0 | match SNMP` command shows that the SNMP ifIndex field has the same value (**503**) as the output for the `show log sampled` command in the "Verifying Sampled Traffic" on page 124 task, indicating that the intended traffic is being sampled.

The output of the `show route 10.4.4.4 detail` command shows that the source address **10.4.4.4**, the source mask (**16**), and the source AS (**65000**) have the same values as the output for the `show log sampled` command in the "Verifying Sampled Traffic" on page 124 task, indicating that the intended traffic is being sampled.

RELATED DOCUMENTATION

Configuring Traffic Sampling on MX, M and T Series Routers | 422

Configuring an Aggregate Export Timer on M, MX and T Series Routers for Version 8 Records

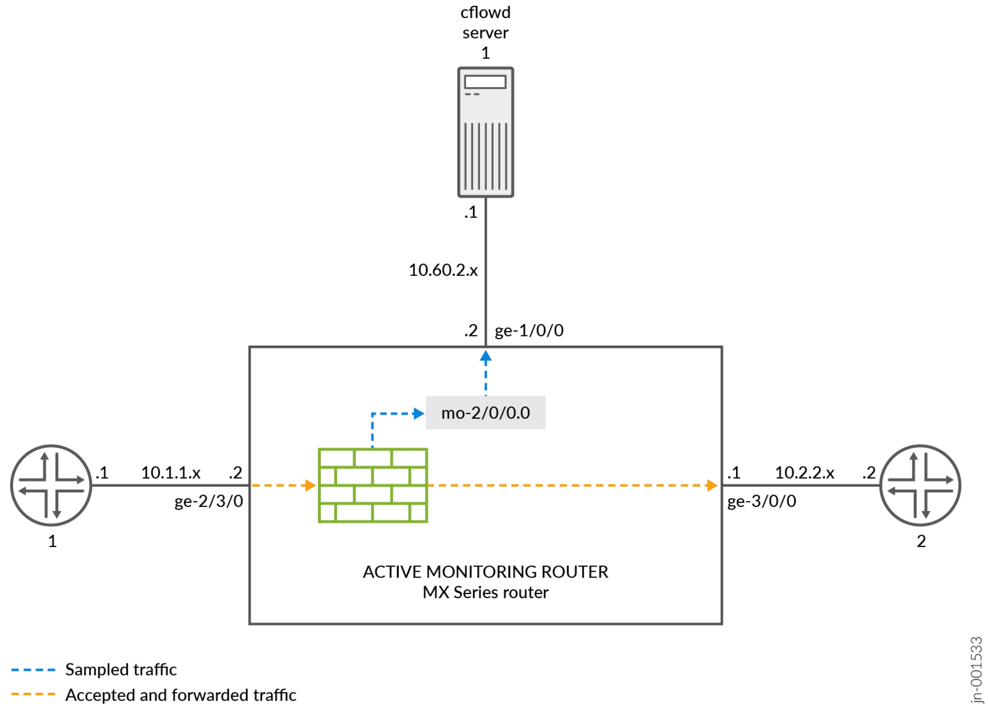
When you use flow monitoring version 8 records for active flow monitoring, you can configure an aggregate export timer. To configure this timer, include the `aggregate-export-interval` statement at the `[edit forwarding-options sampling output]` hierarchy level. The timer value has a default minimum setting of 90 seconds and a maximum value of 1800 seconds.

```
[edit]
forwarding-options {
    sampling {
        output {
            aggregate-export-interval duration;
        }
    }
}
```

Example: Sampling Configuration for MX Series Routers

IN THIS SECTION

Figure 17: Active Flow Monitoring—Sampling Configuration Topology Diagram



In Figure 17 on page 128, traffic from Router 1 arrives on the monitoring router's Gigabit Ethernet **ge-2/3/0** interface. The exit interface on the monitoring router that leads to destination Router 2 is **ge-3/0/0**. In active flow monitoring, both the input interface and exit interface can be any interface type (such as Gigabit Ethernet). The export interface leading to the flow server is **ge-1/0/0**.

Configure a firewall filter to sample, count, and accept all traffic. Apply the filter to the input interface, and configure the exit interface (for traffic forwarding), the adaptive services interface (for flow processing), and the export interface (for exporting flow records).

Configure sampling at the `[edit forwarding-options]` hierarchy level. Include the IP address and port of the flow server with the `flow-server` statement and specify the adaptive services interface to be used for flow record processing with the `interface` statement at the `[edit forwarding-options sampling]` hierarchy level.

Router 1

```
[edit]
interfaces {
   sp-2/0/0 { # This adaptive services interface creates the flow records.
       unit 0 {
           family inet {
               address 10.5.5.1/32 {
```

```
                    destination 10.5.5.2;
                }
            }
        }
    }
    ge-1/0/0 { # This is the interface where records are sent to the flow server.
        unit 0 {
            family inet {
                address 10.60.2.2/30;
            }
        }
    }
    ge-2/3/0 { # This is the input interface where all traffic enters the router.
        unit 0 {
            family inet {
                filter {
                   input catch_all; # This is where the firewall filter is applied.
                }
                address 10.1.1.1/20;
            }
        }
    }
    ge-3/0/0 { # This is the interface where the original traffic is forwarded.
        unit 0 {
            family inet {
                address 10.2.2.1/24;
            }
        }
    }
}
forwarding-options {
   sampling { # Traffic is sampled and sent to a flow server.
        input {
           rate 1; # Samples 1 out of x packets (here, a rate of 1 sample per packet).
        }
        family inet {
            output {
               flow-server 10.60.2.1 { # The IP address and port of the flow server.
                    port 2055;
                    version 5; # Records are sent to the flow server using version 5 format.
                }
                flow-inactive-timeout 15;
                flow-active-timeout 60;
```

```
                interface sp-2/0/0 { # Adding an interface here enables PIC-based sampling.
                    engine-id 5; # Engine statements are dynamic, but can be configured.
                    engine-type 55;
                    source-address 10.60.2.2; # You must configure this statement.
                }
            }
        }
    }
}
firewall {
    family inet {
        filter catch_all { # Apply this filter on the input interface.
            term default {
                then {
                    sample;
                    count counter1;
                    accept;
                }
            }
        }
    }
}
```

Verifying Your Work

To verify that your configuration is correct, use the following commands on the monitoring station that is configured for active flow monitoring:

- `show services accounting errors`
- `show services accounting (flow | flow-detail)`
- `show services accounting memory`
- `show services accounting packet-size-distribution`
- `show services accounting status`
- `show services accounting usage`
- `show services accounting aggregation template template-name` *`name`* `(detail | extensive | terse)` (version 9 only)

Most active flow monitoring operational mode commands contain equivalent output information to the following passive flow monitoring commands:

- `show services accounting errors = show passive-monitoring error`
- `show services accounting flow = show passive-monitoring flow`
- `show services accounting memory = show passive-monitoring memory`
- `show services accounting status = show passive-monitoring status`
- `show services accounting usage = show passive-monitoring usage`

The active flow monitoring commands can be used with most active flow monitoring applications, including sampling, discard accounting, port mirroring, and multiple port mirroring. However, you can use the passive flow monitoring commands only with configurations that contain a monitoring group at the `[edit forwarding-options monitoring]` hierarchy level.

The following shows the output of the `show` commands used with the configuration example:

```
user@router1> show services accounting errors
Service Accounting interface: sp-2/0/0, Local interface index: 542
Service name: (default sampling)
  Error information
    Packets dropped (no memory): 0, Packets dropped (not IP): 0
    Packets dropped (not IPv4): 0, Packets dropped (header too small): 0
    Memory allocation failures: 0, Memory free failures: 0
    Memory free list failures: 0
    Memory overload: No, PPS overload: No, BPS overload: Yes

user@router1> show services accounting flow-detail limit 10
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
Protocol    Source          Source  Destination Destination    Packet        Byte
            Address           Port  Address            Port     count       count
udp(17)     10.1.1.2            53  10.0.0.1             53      4329     3386035
ip(0)       10.1.1.2             0  10.0.0.2              0      4785     3719654
ip(0)       10.1.1.2             0  10.0.1.2              0      4530     3518769
udp(17)     10.1.1.2             0  10.0.7.1              0      5011     3916767
tcp(6)      10.1.1.2            20  10.3.0.1             20         1        1494
tcp(6)      10.1.1.2            20  10.168.80.1          20         1         677
tcp(6)      10.1.1.2            20  10.69.192.1          20         1         446
tcp(6)      10.1.1.2            20  10.239.240.1         20         1        1426
tcp(6)      10.1.1.2            20  10.126.160.1         20         1         889
tcp(6)      10.1.1.2            20  10.71.224.1          20         1        1046

user@router1>  show services accounting memory
```

```
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
  Memory utilization
    Allocation count: 437340, Free count: 430681, Maximum allocated: 6782
    Allocations per second: 3366, Frees per second: 6412
    Total memory used (in bytes): 133416928, Total memory free (in bytes): 133961744

user@router1>  show services accounting packet-size-distribution
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
Range start     Range end     Number of packets     Percentage packets
         64            96               1705156                    100

user@router1>  show services accounting status
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
  Interface state: Monitoring
  Group index: 0
  Export interval: 60 secs, Export format: cflowd v5
  Protocol: IPv4, Engine type: 55, Engine ID: 5
  Route record count: 13, IFL to SNMP index count: 30, AS count: 1
  Time set: Yes, Configuration set: Yes
  Route record set: Yes, IFL SNMP map set: Yes

user@router1>  show services accounting usage
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
  CPU utilization
    Uptime: 4790345 milliseconds, Interrupt time: 1668537848 microseconds
    Load (5 second): 71%, Load (1 minute): 63%
```

Associating Sampling Instances for Active Flow Monitoring with a Specific FPC, MPC, or DPC

The Junos OS enables you to configure sampling instances for active flow monitoring, by specifying a name for the sampling parameters and associating the instance name with a specific FPC, MPC, or DPC.

To configure active sampling instances, include the `instance` statement at the `[edit forwarding-options sampling]` hierarchy level. For more information about configuring sampling instances, see the Junos OS Services Interfaces Library for Routing Devices.

To associate a configured active sampling instance with a specific FPC, MPC, or DPC, include the sampling instance name at the `[edit chassis fpc slot-number]` hierarchy level:

```
[edit chassis fpc slot-number]
sampling-instance instance-name;
```

RELATED DOCUMENTATION

Example: Sampling Instance Configuration | 133

sampling-instance

Example: Sampling Instance Configuration

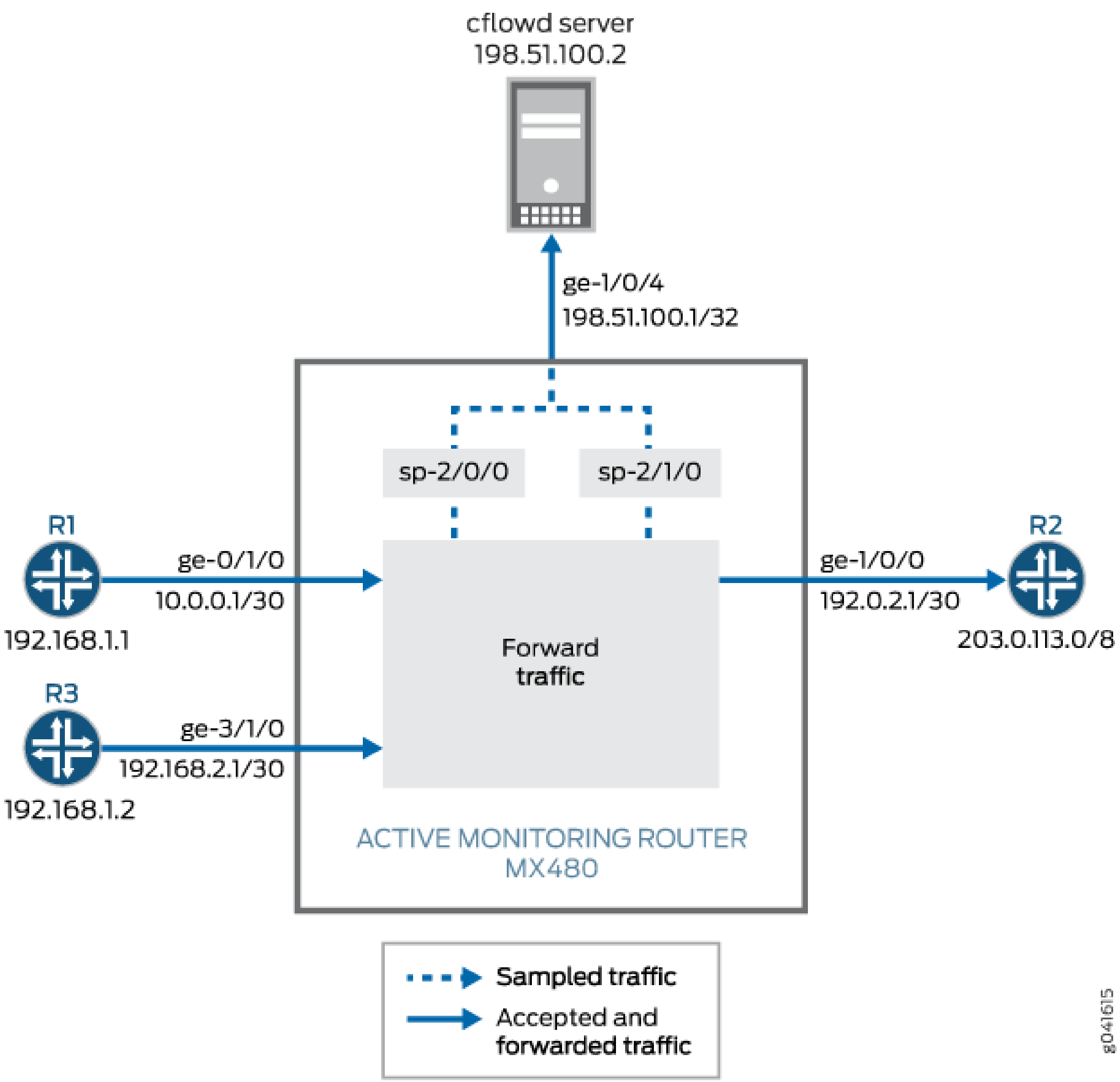
IN THIS SECTION

You can configure active sampling using a sampling instance and associate that sampling instance to a particular Flexible Port Concentrator (FPC), Modular Port Concentrator (MPC), or Dense Port Concentrator (DPC). In addition, you can define multiple sampling instances associated with multiple destinations and protocol families per sampling instance destination.

Example Network Details

The following example shows the configuration of two sampling instances on an MX480 router running Junos OS Release 9.6.

Figure 18: Active Flow Monitoring—Sampling Instance Configuration Topology Diagram



In Figure 18 on page 134, packets from Router 1 arrive on the monitoring router's Gigabit Ethernet `ge-0/1/0` interface, the packets are sampled by the services interface `sp-2/0/0` and sent to the cflowd server by the export interface `ge-1/0/4`. Packets from Router 3 arrive on the monitoring router's Gigabit Ethernet `ge-3/1/0` interface, the packets are sampled by the services interface `sp-2/1/0` and sent to the cflowd server by the export interface `ge-1/0/4`. Normal traffic flow from `ge-0/1/0` and `ge-3/1/0` to `ge-1/0/0` and on to Router 2 continues undisturbed during the sampling process. In active flow monitoring, both the input interface and exit interface can be any interface type (such as SONET/SDH, Gigabit Ethernet, and so on).

Only one sampling instance can be attached to an FPC, MPC, or DPC. Multiple families can be configured under a sampling instance. Each family can have its own collector address. You can define sampling instances and attach each instance to different FPCs, or a single sampling instance can be attached to all FPCs.

The sampling configuration for this example includes the following:

- Two sampling instances, `s0` and `s1`, configured to collect sampling data at the `[edit forwarding-options]` hierarchy level. The `flow-server` statement includes the IP address, port, and template of the flow server. The `interface` statement includes the services interface, `sp-2/0/0` or `sp-2/1/0`, for flow record processing, and the source address of the incoming router on the sampled interface.

- The binding of the two sampling instances to FPCs 0 and 3. These are configured with the `sampling-instance` statement at the `[edit chassis fpc` *`slot`*`]` hierarchy level.

- Sampling activated on the input interfaces `ge-0/1/0` and `ge-3/1/0` using the `sampling` statement at the `[edit interfaces` *`interface-name`* `unit` *`unit-number`* `family` *`family`*`]` hierarchy level.

In this example, the `ping` command is issued on Router 1 to Router 2 via the MX480 router to generate traffic. After the packets are generated, `show` commands are issued to verify that the sampling configuration is working as expected.

Example Router Configuration

The following output shows the configuration of an MX480 router with two sampling instances.

```
user@MX480-router> show configuration
[...Output Truncated...]
}
chassis {
    fpc 0 { # The fpc number is associated with the interface on which sampling is enabled,
ge-0/1/0 in this statement.
        sampling-instance s0;
    }
    fpc 3 { # The fpc number is associated with the interface on which sampling is enabled,
ge-3/1/0 in this statement.
        sampling-instance s1;
    }
}
interfaces {
    ge-0/1/0 { # This interface has sampling activated.
        unit 0 {
            family inet {
                sampling { # Here sampling is activated.
                    input;
```

```
                }
                address 10.0.0.1/30;
            }
        }
    }
    ge-1/0/0 { # The interface on which packets are exiting the router.
        unit 0 {
            family inet {
                address 192.0.2.1/30;
            }
        }
    }
    ge-1/0/4 { # The interface connected to the cflowd server.
        unit 0 {
            family inet {
                address 198.51.100.1/32;
            }
        }
    }
    sp-2/0/0 { # The service interface that samples the packets from Router 1.
        unit 0 {
            family inet;
        }
    }
    sp-2/1/0 { # The service interface that samples the packets from Router 3.
        unit 0 {
            family inet;
        }
    }
    ge-3/1/0 { # This interface has sampling activated.
        unit 0 {
            family inet {
                sampling { # Here sampling is activated.
                    input;
                }
                address 192.168.2.1/30;
            }
        }
    }
}
forwarding-options {
    sampling {
        instance {
```

```
        s0 {
            input {
                rate 1;
                run-length 0;
            }
            family inet {
                output {
                    flow-server 198.51.100.2 { # The address of the external server.
                        port 2055;
                        version9 {
                            template {
                                v4
                            }
                        }
                    }
                    interface sp-2/0/0 {
                        source-address 192.168.1.1; # Source address of the sampled packets
                    }
                }
            }
        }
        s1 {
            input {
                rate 1;
                run-length 0;
            }
            family inet {
                output {
                    flow-server 198.51.100.2 { # The address of the external server.
                        port 2055;
                        version9 {
                            template {
                                v4
                            }
                        }
                    }
                    interface sp-2/1/0 {
                        source-address 192.168.1.2; # Source address of the sampled packets
                    }
                }
            }
        }
    }
```

```
    }
}

routing-options {
    static {
        route 203.0.113.0/8 next-hop 192.0.2.2;
    }
}
services {
    flow-monitoring {
        version9 {
            template v4 {
                flow-active-timeout 30;
                flow-inactive-timeout 30;
                ipv4-template;
            }
        }
    }
}
```

Configuration Commands Used for the Configuration Example

The following `set` commands are used for the configuration of the sampling instance in this example. Replace the values in these commands with values relevant to your own network.

- `set chassis fpc 0 sampling-instance s0`
- `set chassis fpc 3 sampling-instance s1`
- `set interfaces ge-0/1/0 unit 0 family inet sampling input`
- `set interfaces ge-0/1/0 unit 0 family inet address`
- `set interfaces ge-1/0/0 unit 0 family inet address`
- `set interfaces sp-2/0/0 unit 0 family inet`
- `set interfaces sp-2/1/0 unit 0 family inet`
- `set interfaces ge-3/1/0 unit 0 family inet sampling input`
- `set interfaces ge-3/1/0 unit 0 family inet address`
- `set forwarding-options sampling instance s0 input rate 1`

- `set forwarding-options sampling instance s0 input run-length 0`
- `set forwarding-options sampling instance s0 family inet output flow-server 198.51.100.2 port 2055`
- `set forwarding-options sampling instance s0 family inet output flow-server 198.51.100.2 version9 template v4;`
- `set forwarding-options sampling instance s0 family inet output interface sp-2/0/0 source-address 192.168.1.1`
- `set forwarding-options sampling instance s1 input rate 1`
- `set forwarding-options sampling instance s1 input run-length 0`
- `set forwarding-options sampling instance s1 family inet output flow-server 198.51.100.2 port 2055`
- `set forwarding-options sampling instance s1 family inet output flow-server 198.51.100.2 version9 template v4;`
- `set forwarding-options sampling instance s1 family inet output interface sp-2/1/0 source-address 192.168.1.2`
- `set routing-options static route 203.0.113.0/8 next-hop 192.0.2.2`
- `set services flow-monitoring version9 template v4 flow-active-timeout 30`
- `set services flow-monitoring version9 template v4 flow-inactive-timeout 30`
- `set services flow-monitoring version9 template v4 ipv4-template`

Verifying Your Work

To verify that your configuration is working as expected, use the following commands on the router that is configured with the sampling instance:

- `show services accounting aggregation template template-name` *`template-name`*
- `show services accounting flow`

The following shows the output of the `show` commands issued on the MX480 router used in this configuration example:

```
user@MX480-router> show services accounting aggregation template template-name v4
                                Src   Dst
                              Port/ Port/
Source          Destination     ICMP  ICMP                         Packet
Address         Address         Type  Code  Proto TOS               Count
10.0.0.6        203.0.113.3      100  1000     17   8                  14
10.0.0.5        203.0.113.2      100  1000     17   8                  15
10.0.0.3        203.0.113.3      100  1000     17   8                  15
10.0.0.2        203.0.113.3      100  1000     17   8                  15
```

```
10.0.0.4        203.0.113.2       100  1000     17   8                   15
10.0.0.6        203.0.113.2       100  1000     17   8                   15
10.0.0.4        203.0.113.3       100  1000     17   8                   15
10.0.0.2        203.0.113.2       100  1000     17   8                   16
10.0.0.3        203.0.113.2       100  1000     17   8                   15
10.0.0.5        203.0.113.3       100  1000     17   8                   15

user@MX480-router> show services accounting aggregation template template-name v4
                                  Src   Dst
                                Port/ Port/
Source          Destination      ICMP  ICMP                          Packet
Address         Address          Type  Code  Proto TOS                Count
10.0.0.6        203.0.113.3      100  1000     17   8                  16
10.0.0.5        203.0.113.2      100  1000     17   8                  17
10.0.0.3        203.0.113.3      100  1000     17   8                  16
10.0.0.2        203.0.113.3      100  1000     17   8                  16
10.0.0.4        203.0.113.2      100  1000     17   8                  17
10.0.0.6        203.0.113.2      100  1000     17   8                  17
10.0.0.4        203.0.113.3      100  1000     17   8                  16
10.0.0.2        203.0.113.2      100  1000     17   8                  17
10.0.0.3        203.0.113.2      100  1000     17   8                  17
10.0.0.5        203.0.113.3      100  1000     17   8                  16

user@MX480-router> show services accounting flow
  Flow information
    Interface name: sp-2/0/0, Local interface index: 152
    Flow packets: 884, Flow bytes: 56576
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 628
    Active flows: 10, Total flows: 35
    Flows exported: 75, Flows packets exported: 14
    Flows inactive timed out: 25, Flows active timed out: 75

user@MX480-router> show services accounting flow
  Flow information
    Interface name: sp-2/0/0, Local interface index: 152
    Flow packets: 898, Flow bytes: 57472
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 628
    Active flows: 10, Total flows: 35
    Flows exported: 75, Flows packets exported: 14
    Flows inactive timed out: 25, Flows active timed out: 75
```

RELATED DOCUMENTATION

Configuring Sampling Instance on MX, M and T Series Routers or QFX Series Switches | 435

Configuring Active Flow Monitoring | 42

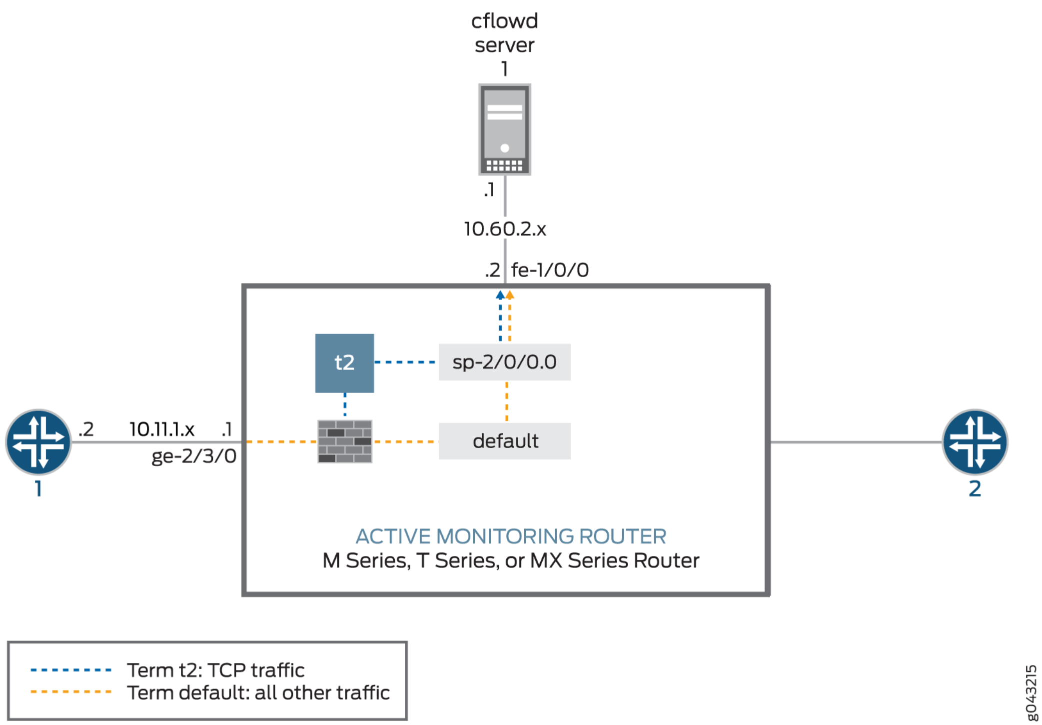
sampling-instance

Example: Sampling and Discard Accounting Configuration on M, MX and T Series Routers

IN THIS SECTION

Discard accounting allows you to sample traffic, send it to a flow server for analysis, and discard all packets without forwarding them to their intended destination. Discard accounting is enabled with the `discard accounting` *`group-name`* statement in a firewall filter at the `[edit firewall family inet filter` *`filter-name`* `term` *`term-name`* `then]` hierarchy level. Then, the filter is applied to an interface with the `filter` statement at the `[edit interfaces` *`interface-name`* `unit` *`unit-number`* `family inet]` hierarchy level and processed with the `output` statement at the `[edit forwarding-options accounting` *`group-name`*`]` hierarchy level.

Figure 19: Active Flow Monitoring—Sampling and Discard Accounting Topology Diagram



In Figure 19 on page 142, traffic from Router 1 arrives on the monitoring router's Gigabit Ethernet **ge-2/3/0** interface. The export interface leading to the flow server is **fe-1/0/0** and there is no exit interface.

In this example, TCP traffic is sent to one accounting group and all other traffic is diverted to a second group. After being sampled and counted, the two types of traffic are acted upon by the sampling and accounting processes. These processes create flow records and send the records to the version 8 flow server for analysis. Because multiple types of traffic are sent to the same server, we recommend that you configure the **engine-id**, **engine-type**, and `source-address` statements manually in your accounting and sampling hierarchies. This way, you can differentiate between traffic types when they arrive at the flow server.

```
[edit]
interfaces {
    sp-2/0/0 { # This adaptive services interface creates the flow records.
        unit 0 {
            family inet {
                address 10.5.5.1/32 {
                    destination 10.5.5.2;
                }
```

```
            }
        }
    }
                   fe-1/0/0 { # This is the interface where records are sent to the flow server.
        unit 0 {
            family inet {
                address 10.60.2.2/30;
            }
        }
    }
                   ge-2/3/0 { # This is the input interface where traffic enters the router.
        unit 0 {
            family inet {
                filter {
                    input catch_all;
                }
                address 10.11.1.1/30;
            }
        }
    }
}
forwarding-options {
                   sampling { # The router samples the traffic.
        input {
                               rate 100; # One out of every 100 packets is sampled.
            }
        }
            family inet {
                                 output { # The sampling process creates and exports flow
records.
                                   flow-server 10.60.2.1 { # You can configure a variety of
settings.
                    port 2055;
                    version 8;
                                         aggregation { # Aggregation is unique to flow version 8.
                        protocol-port;
                        source-destination-prefix;
                    }
                }
                aggregate-export-interval 90;
                flow-inactive-timeout 60;
                flow-active-timeout 60;
                                   interface sp-2/0/0 { # This statement enables PIC-based
```

```
sampling.
                    engine-id 5; # Engine statements are dynamic, but can be configured.
                    engine-type 55;
                                       source-address 10.60.2.2; # You must configure this
statement.
                }
            }
    }
                   accounting counter1 { # This discard accounting process handles default
traffic.
                         output { # This process creates and exports flow records.
            flow-inactive-timeout 65;
            flow-active-timeout 65;
                               flow-server 10.60.2.1 { # You can configure a variety of settings.
                port 2055;
                version 8;
                                     aggregation { # Aggregation is unique to version 8.
                    protocol-port;
                    source-destination-prefix;
                }
            }
                               interface sp-2/0/0 {  # This statement enables PIC-based discard
accounting.
                engine-id 1; # Engine statements are dynamic, but can be configured.
                engine-type 11;
                                   source-address 10.60.2.3; # You must configure this statement.
            }
        }
    }
                   accounting t2 { # The second discard accounting process handles the TCP
traffic.
                         output { # This process creates and exports flow records.
            aggregate-export-interval 90;
            flow-inactive-timeout 65;
            flow-active-timeout 65;
                               flow-server 10.60.2.1 {                  # You can configure a
variety of settings for the server.
                port 2055;
                version 8;
                                     aggregation { # Aggregation is unique to version 8.
                    protocol-port;
                    source-destination-prefix;
                }
```

```
            }
                                interface sp-2/0/0 { # This statement enables PIC-based discard
accounting.
                engine-id 2; # Engine statements are dynamic, but can be configured.
                engine-type 22;
                                    source-address 10.60.2.4;# You must configure this statement.
            }
        }
    }
}
firewall {
    family inet {
                          filter catch_all { # Apply the firewall filter on the input interface.
                                term t2 { # This places TCP traffic into one group for sampling
and
                from { # discard accounting.
                    protocol tcp;
                }
                then {
                                          count c2;# The count action counts traffic as it enters
the router.
                                          sample; # The sample action sends the traffic to the
sampling process.
                                          discard accounting t2; # The discard accounting
discards traffic.
                }
            }
                                term default { # Performs sampling and discard accounting on all
other traffic.
                then {
                                          count counter; # The count action counts traffic as it
enters the router.
                                          sample# The sample action sends the traffic to the
sampling process.
                                          discard accounting counter1; # This activates discard
accounting.
                }
            }
        }
    }
}
```

Verifying Your Work

To verify that your configuration is correct, use the following commands on the monitoring station that is configured for active flow monitoring:

- `show services accounting aggregation` (for version 8 flows only)
- `show services accounting errors`
- `show services accounting (flow | flow-detail)`
- `show services accounting memory`
- `show services accounting packet-size-distribution`
- `show services accounting status`
- `show services accounting usage`

The following shows the output of the `show` commands used with the configuration example:

```
user@host> show services accounting flow name t2
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: t2
  Flow information
    Flow packets: 56130820, Flow bytes: 3592372480
    Flow packets 10-second rate: 13024, Flow bytes 10-second rate: 833573
    Active flows: 600, Total flows: 600
    Flows exported: 28848, Flows packets exported: 960
    Flows inactive timed out: 0, Flows active timed out: 35400

user@host>  show services accounting
Service Name:
    (default sampling)
    counter1
    t2

user@host>  show services accounting aggregation protocol-port detail name t2
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: t2

  Protocol: 6, Source port: 20, Destination port: 20
  Start time: 442794, End time: 6436260
  Flow count: 1, Packet count: 4294693925, Byte count: 4277471552
```

```
user@host>  show services accounting aggregation source-destination-prefix name
t2 limit 10 order packets
Service Accounting interface: sp-2/0/0, Local interface index: 542
Service name: t2
     Source     Destination  Input SNMP  Output SNMP   Flow  Packet    Byte
     Prefix          Prefix       Index        Index  count   count   count
10.1.1.2/20    10.225.0.1/0          24           26      0      13    9650
10.1.1.2/20   10.143.80.1/0          24           26      0      13   10061
10.1.1.2/20   10.59.176.1/0          24           26      0      13   10426
10.1.1.2/20     10.5.32.1/0          24           26      0      13   12225
10.1.1.2/20    10.36.16.1/0          24           26      0      13    9116
10.1.1.2/20     10.1.96.1/0          24           26      0      12   11050
10.1.1.2/20    10.14.48.1/0          24           26      0      13   10812
10.1.1.2/20   10.31.192.1/0          24           26      0      13   11473
10.1.1.2/20  10.129.144.1/0          24           26      0      13    7647
10.1.1.2/20  10.188.160.1/0          24           26      0      13   10056

user@host> show services accounting aggregation source-destination-prefix name
t2 extensive limit 3
Service Accounting interface: sp-2/0/0, Local interface index: 542
Service name: t2

  Source address: 10.1.1.2, Source prefix length: 20
  Destination address: 10.200.176.1, Destination prefix length: 0
  Input SNMP interface index: 24, Output SNMP interface index: 26
  Source-AS: 69, Destination-AS: 69
  Start time: Fri Feb 21 14:16:57 2003, End time: Fri Feb 21 14:22:50 2003
  Flow count: 0, Packet count: 6, Byte count: 5340

  Source address: 10.1.1.2, Source prefix length: 20
  Destination address: 10.243.160.1, Destination prefix length: 0
  Input SNMP interface index: 24, Output SNMP interface index: 26
  Source-AS: 69, Destination-AS: 69
  Start time: Fri Feb 21 14:16:57 2003, End time: Fri Feb 21 14:22:50 2003
  Flow count: 0, Packet count: 6, Byte count: 5490

  Source address: 10.1.1.2, Source prefix length: 20
  Destination address: 10.162.160.1, Destination prefix length: 0
  Input SNMP interface index: 24, Output SNMP interface index: 26
  Source-AS: 69, Destination-AS: 69
```

```
Start time: Fri Feb 21 14:16:57 2003, End time: Fri Feb 21 14:22:50 2003
Flow count: 0, Packet count: 6, Byte count: 4079
```

CHAPTER 3

Monitoring Traffic Using Passive Flow Monitoring

IN THIS CHAPTER

Passive Flow Monitoring Overview

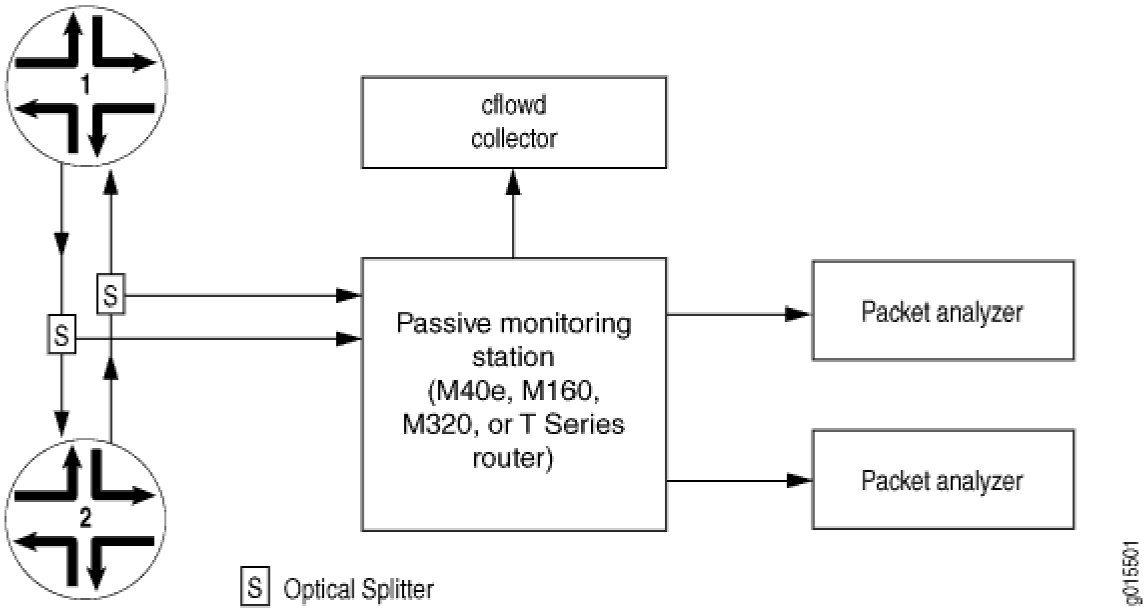
Using an MX Series router, a selection of PICs (including the Monitoring Services PIC, Adaptive Services [AS] PIC, Multiservices PIC, or Multiservices DPC) and other networking hardware, you can monitor traffic flow and export the monitored traffic. Monitoring traffic allows you to do the following:

- Gather and export detailed information about IP version 4 (IPv4) traffic flows between source and destination nodes in your network.
- Sample all incoming IPv4 traffic on the monitoring interface and present the data in cflowd record format.
- Perform discard accounting on an incoming traffic flow.
- Encrypt or tunnel outgoing cflowd records, intercepted IPv4 traffic, or both.
- Direct filtered traffic to different packet analyzers and present the data in its original format (port mirror).

> **NOTE**: Monitoring Services PICs, AS PICs, and Multiservices PICs must be mounted on an Enhanced Flexible PIC Concentrator (FPC) in an MX Series router.
>
> Multiservices DPCs installed in Juniper Networks MX Series routers support the same functionality, with the exception of the passive monitoring and flow-tap features.

The router used for passive monitoring does not route packets from the monitored interface, nor does it run any routing protocols related to those interfaces; it only receives traffic flows, collects intercepted traffic, and exports it to cflowd servers and packet analyzers. Figure 20 on page 151 shows a typical topology for the passive flow-monitoring application.

Figure 20: Passive Monitoring Application Topology



Traffic travels normally between Router 1 and Router 2. To redirect IPv4 traffic, you insert an optical splitter on the interface between these two routers. The optical splitter copies and redirects the traffic to the monitoring station, which is an MX Series router. The optical cable connects only the receive port on the monitoring station, never the transmit port. This configuration allows the monitoring station to receive traffic from the router being monitored but never to transmit it back.

If you are monitoring traffic flow, the Internet Processor II application-specific integrated circuit (ASIC) in the router forwards a copy of the traffic to the Monitoring Services, Adaptive Services, or Multiservices PIC in the monitoring station. If more than one monitoring PIC is installed, the monitoring station distributes the load of the incoming traffic across the multiple PICs. The monitoring PICs generate flow records in cflowd version 5 format, and the records are then exported to the cflowd collector.

If you are performing lawful interception of traffic between the two routers, the Internet Processor II ASIC filters the incoming traffic and forwards it to the Tunnel Services PIC. Filter-based forwarding is then applied to direct the traffic to the packet analyzers.

Optionally, the intercepted traffic or the cflowd records can be encrypted by the ES PIC or IP Security (IPsec) services and then sent to a cflowd server or packet analyzer.

RELATED DOCUMENTATION

Enabling Passive Flow Monitoring on M Series, MX Series or T Series Routers | 156

Passive Flow Monitoring System Requirements for MX Series Routers

To perform passive flow monitoring, your router must meet these minimum requirements:

- Junos OS Release 22.4R1 or later for passive flow monitoring support on the MX304 router with the LMIC16-BASE line card, on the MX10004, MX10008, and MX10016 routers with the LC9600 line card and on the MX2010 and MX2020 routers with the MPC10 and MPC11 line cards.
- Junos OS Release 20.4R1 or later for passive flow monitoring support on the MX10008 router with the JNP10K-2101 line card and on the MX240/MX480/MX960/MX2008/MX2010/MX2020 routers with either the MPC7E-MRATE or MPC7E-10G line card.
-
- Type 1 enhanced FPCs
- Two optical splitters
- A Tunnel Services PIC (required if you wish to send traffic to more than one analyzer)
- An input interface from the following list:
  - Gigabit Ethernet PIC—4-port with small form-factor pluggable transceiver (SFP) or 10-port with SFP
  - 1-port 10-Gigabit Ethernet PIC with XENPAK
- Outgoing PICs to connect to the flow collector or packet analyzer
- Flow monitoring version 5 collector
- ES PIC and packet analyzers (optional)

RELATED DOCUMENTATION

Active Flow Monitoring System Requirements | 45

Active Flow Monitoring PIC Specifications | 48

Passive Flow Monitoring Router and Software Considerations for MX Series Routers

There are several hardware and software considerations when you implement passive flow monitoring. When defining the hardware requirements of the monitoring station, keep in mind the following:

- The input interfaces on the monitoring station must be Gigabit Ethernet interfaces with SFP (4-port or 10-port) or 1-port 10-Gigabit Ethernet interfaces with XENPAK.
- To monitor the flows in both directions for a single interface, the monitoring station must have two Ethernet-based receive ports, one for each direction of flow. In "Passive Flow Monitoring Application Topology" on page 154, the monitoring station needs one port to monitor the traffic flowing from Router 1 to Router 2, and a second port to monitor the traffic flowing from Router 2 to Router 1.
- The Monitoring Services PICs must be installed in a Type 1 enhanced FPC slot.
- Type 1 and Type 2 Tunnel Services PICs are supported.
- Use an ES PIC to encrypt the flow export.
- Symmetric hashing is not supported on the MPC10 and MPC11 line cards. You should choose a different MPC line card if you wish to support symmetrical hashing along with passive monitoring.
- You can only configure passive monitoring on a physical port and not on a logical interface or per VLAN. You cannot configure passive monitoring on an aggregated Ethernet port or on a port with Ethernet encapsulation.
- IDS servers must be directly connected to the router. You need to configure the interfaces connecting to the IDS servers as part of a link aggregation group (LAG). You need to configure static routes to route the packets onto an IDS server.

When defining a traffic monitoring strategy, keep in mind the following:

- The monitoring station collects only IPv4 packets. All other packet formats are discarded and not counted.
- You can set the amount of time a data flow can be inactive before the monitoring station terminates the flow and exports the flow data. To set the timer, include the `flow-inactive-timeout` statement at the `[edit forwarding-options monitoring` *`group-name`* `family inet output]` hierarchy level. The timer value can be from 15 seconds through 1800 seconds, with a default value of 60 seconds.

You can also configure the monitoring station to collect periodic flow reports for flows that last longer than the configured active timeout. To set this activity timer, include the `flow-active-timeout` statement at the `[edit forwarding-options monitoring` *`group-name`* `family inet output]` hierarchy level. The timer value can be from 60 seconds through 1800 seconds, with a default value of 180 seconds.

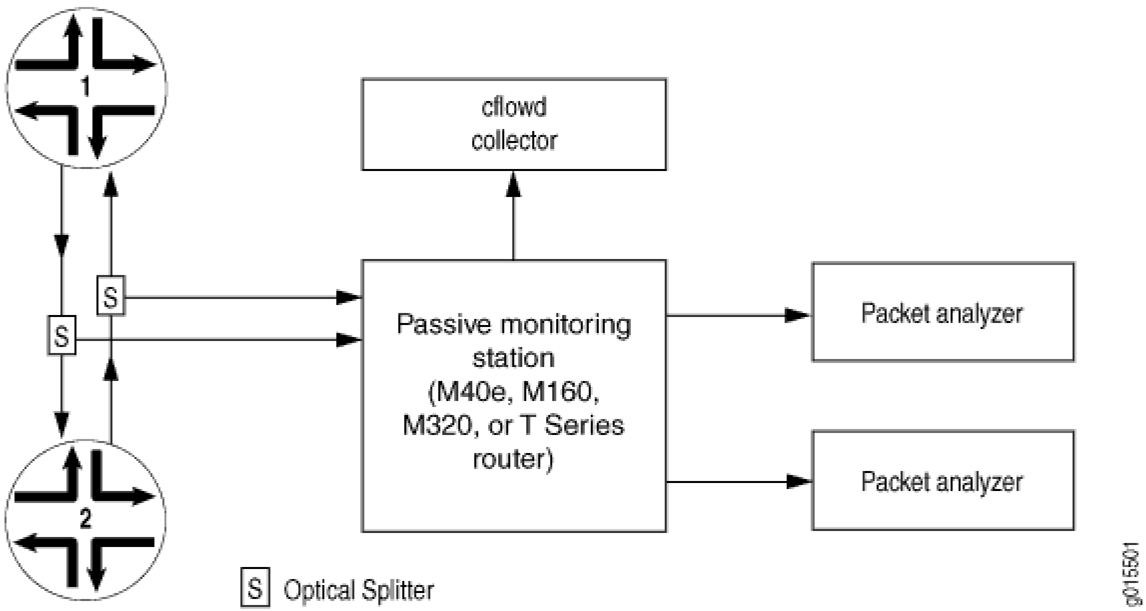
- Multiple expired flows are exported together, if possible. A UDP packet is sent when one of the following conditions is met:
  - When 30 flows are contained in the current packet, the flows are exported.
  - If there are fewer than 30 flows but the export timer expires, the flows are exported one second after the timer expires.
- TCP and UDP flows are considered differently:
  - TCP flows watch for a segment containing the **FIN** bit and a subsequent acknowledgement (**ACK**) to detect the end of a flow. Alternately, a TCP reset (**RST**) can also indicate the end of a flow. When these TCP combinations are detected, the flow expires. The **FIN+ACK** and **RST** cases cover most TCP stream closures. For all other flows, an inactive timeout is needed.
  - All non-TCP flows, such as UDP, depend on timeout mechanisms for export.
- The default MTU value for Gigabit Ethernet interfaces is 1500 bytes. If the monitoring station receives packets exceeding 4474 bytes, they are discarded; no fragmentation is performed. Note that the supported MTU size on the Gigabit Ethernet PICs might exceed 1500 bytes, depending on the type of PIC.
- Any incoming traffic that is discarded is not forwarded to packet analyzers.
- The interfaces on the monitoring station that collect intercepted traffic must be configured with Cisco HDLC or PPP encapsulation.
- You must always use a standard interface (for example, one that follows the usual ***interface-name-fpc/pic/slot*** format) to send flow records to a flow server. Flow data generated by the Monitoring Services or Monitoring Services II PICs will not be delivered to the server across the **fxp0** interface.
- You can send version 5 records to multiple flow servers. You can configure up to eight servers and flow traffic is load-balanced between the servers in a round-robin fashion. If one of the servers ceases operation, flow traffic load-balances automatically between the remaining active servers. To configure, include up to eight `flow-server` statements at the `[edit forwarding-options monitoring *group-name* output]` hierarchy level.

Understanding Passive Flow Monitoring on MX Series Routers

Flow monitoring version 5 supports passive flow monitoring. Versions 8 and 9 do not support passive flow monitoring.

The MX Series router that is used for passive flow monitoring does not route packets from monitored interfaces, nor does it run any routing protocols related to those interfaces; it only passes along intercepted traffic and receives traffic flows. Figure 21 on page 155 shows a typical topology for the passive flow monitoring application.

Figure 21: Passive Flow Monitoring Application Topology



Traffic travels normally between Router 1 and Router 2. To redirect IPv4 traffic, you insert an optical splitter on the interface between these two routers. The optical splitter copies and redirects the traffic to the monitoring station. The optical cable connects only the receive port on the monitoring station, never the transmit port. This configuration allows the monitoring station to receive traffic only from the router being monitored but never to transmit it back.

If you are monitoring traffic flow, the Internet Processor II ASIC in the router forwards a copy of the traffic to the Monitoring Services or Monitoring Services II PIC in the monitoring station. If there is more than one Monitoring Services PIC installed, the monitoring station distributes the load of the incoming traffic across the multiple PICs. The Monitoring Services PICs generate flow records in version 5 format, and the records are exported to the flow collector.

When you are performing lawful interception of packets, the Internet Processor II ASIC filters the incoming traffic and forwards it to the Tunnel Services PIC. Filter-based forwarding is then applied to direct the traffic to the packet analyzers. Optionally, the intercepted traffic or the flow records can be encrypted by the ES PIC and then sent to their destination. With additional configuration, flow records can be processed by a flow collector and flows can be captured dynamically.

With MPLS passive monitoring, the router can process MPLS packets with label values that do not have corresponding entries in the `mpls.0` routing table. You can divert these unrecognized MPLS packets, remove the MPLS labels, and redirect the underlying IPv4 packets. This is equivalent to a default route for MPLS packets or a promiscuous label. Because this application does not use a Monitoring Services

PIC, see the *Junos MPLS Applications Configuration Guide* for more information about MPLS passive monitoring.

RELATED DOCUMENTATION

Active Flow Monitoring Overview | 54
Active Flow Monitoring Overview | 53

Enabling Passive Flow Monitoring on M Series, MX Series or T Series Routers

IN THIS SECTION

You can monitor IPv4 traffic from another router if you have the following components installed in an M Series, MX Series, or T Series router:

- Monitoring Services, Adaptive Services, or Multiservices PICs to perform the service processing
- SONET/SDH, Fast Ethernet, or Gigabit Ethernet PICs as transit interface

On SONET/SDH interfaces, you enable passive flow monitoring by including the `passive-monitor-mode` statement at the [edit interfaces so-*fpc*/*pic*/*port* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces so-fpc/pic/port unit logical-unit-number]
passive-monitor-mode;
```

On Asynchronous Transfer Mode (ATM), Fast Ethernet, or Gigabit Ethernet interfaces, you enable passive flow monitoring by including the `passive-monitor-mode` statement at the `[edit interfaces *interface-name*]` hierarchy level:

```
[edit interfaces interface-name]
passive-monitor-mode;
```

IPv6 passive monitoring is not supported on Monitoring Services PICs. You must configure port mirroring to forward the packets from the passive monitored ports to other interfaces. Interfaces configured on the following FPCs and PIC support IPv6 passive monitoring on the T640 and T1600 Series routers:

- Enhanced Scaling FPC2
- Enhanced Scaling FPC3
- Enhanced II FPC1
- Enhanced II FPC2
- Enhanced II FPC3
- Enhanced Scaling FPC4
- Enhanced Scaling FPC4.1
- 4-port 10-Gigabit Ethernet LAN/WAN PIC with XFP (supported on both WAN-PHY and LAN-PHY mode for both IPv4 and IPv6 addresses)
- Gigabit Ethernet PIC with SFP
- 10-Gigabit Ethernet PIC with XENPAK (T1600 Series router)
- SONET/SDH OC192/STM64 PIC (T1600 Series router)
- SONET/SDH OC192/STM64 PICs with XFP (T1600 Series router)
- SONET/SDH OC48c/STM16 PIC with SFP (T1600 Series router)
- SONET/SDH OC48/STM16 (Multi-Rate)
- SONET/SDH OC12/STM4 (Multi–Rate) PIC with SFP
- Type 1 SONET/SDH OC3/STM1 (Multi–Rate) PIC with SFP

To configure port mirroring, include the `port-mirroring` statement at the `[edit forwarding-options]` hierarchy level.

When you configure an interface in passive monitoring mode, the Packet Forwarding Engine silently drops packets coming from that interface and destined to the router itself. Passive monitoring mode also stops the Routing Engine from transmitting any packet from that interface. Packets received from the monitored interface can be forwarded to monitoring interfaces. If you include the `passive-monitor-mode` statement in the configuration:

- The ATM interface is always up, and the interface does not receive or transmit incoming control packets, such as Operation, Administration, and Maintenance (OAM) and Interim Local Management Interface (ILMI) cells.
- The SONET/SDH interface does not send keepalives or alarms and does not participate actively on the network.
- Gigabit and Fast Ethernet interfaces can support both per-port passive monitoring and per-VLAN passive monitoring. The destination MAC filter on the receive port of the Ethernet interfaces is disabled.
- Ethernet encapsulation options are not allowed.
- Ethernet interfaces do not support the `stacked-vlan-tagging` statement for both IPv4 and IPv6 packets in passive monitoring mode.

On monitoring services interfaces, you enable passive flow monitoring by including the `family` statement at the `[edit interfaces *interface-name* unit *logical-unit-number*]` hierarchy level, specifying the `inet` option:

```
[edit interfaces interface-name unit logical-unit-number]
family inet;
```

For the monitoring services interface, you can configure multiservice physical interface properties. For more information, see "Configuring Flow-Monitoring Interfaces" on page 5.

For conformity with the cflowd record structure, you must include the `receive-options-packets` and `receive-ttl-exceeded` statements at the `[edit interfaces *interface-name* unit *logical-unit-number* family inet]` hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet]
receive-options-packets;
receive-ttl-exceeded;
```

Passive Flow Monitoring for MPLS Encapsulated Packets

On monitoring services interfaces, you can process MPLS packets that have not been assigned label values and have no corresponding entry in the `mpls.0` routing table. This allows you to assign a default route to unlabeled MPLS packets.

To configure a default label value for MPLS packets, include the `default-route` statement at the `[edit protocols mpls interface interface-name label-map]` hierarchy level:

```
[edit protocols mpls interface interface-name label-map]
default-route {
    (next-hop (address | interface-name | address/interface-name)) | (reject | discard);
    (pop | (swap <out-label>);
    class-of-service value;
    preference preference;
    type type;
}
```

For more information about static labels, see the MPLS Applications User Guide.

Removing MPLS Labels from Incoming Packets

The Junos OS can forward only IPv4 packets to a Monitoring Services, Adaptive Services, or Multiservices PIC. IPv4 and IPv6 packets with MPLS labels cannot be forwarded to a monitoring PIC. By default, if packets with MPLS labels are forwarded to the monitoring PIC, they are discarded. To monitor IPv4 and IPv6 packets with MPLS labels, you must remove the MPLS labels as the packets arrive on the interface.

You can remove MPLS labels from an incoming packet by including the `pop-all-labels` statement at the `[edit interfaces interface-name (atm-options | fastether-options | gigether-options | sonet-options) mpls]` hierarchy level:

```
[edit interfaces interface-name (atm-options | fastether-options | gigether-options | sonet-
options) mpls]
pop-all-labels {
    required-depth [ numbers ];
}
```

For MX Series routers with MPCs, the `pop-all-labels` statement pops all labels by default and the `required-depth` statement is ignored.

For other configurations, you can remove up to two MPLS labels from an incoming packet. By default, the `pop-all-labels` statement takes effect for incoming packets with one or two labels. You can specify the number of MPLS labels that an incoming packet must have for the `pop-all-labels` statement to take effect by including the `required-depth` statement at the `[edit interfaces interface-name (atm-options | fastether-options | gigether-options | sonet-options) mpls pop-all-labels]` hierarchy level:

```
[edit interfaces interface-name (atm-options | fastether-options | gigether-options | sonet-
options) mpls pop-all-labels]
required-depth [ numbers ];
```

The required depth can be `1`, `2`, or `[ 1 2 ]`. If you include the `required-depth 1` statement, the `pop-all-labels` statement takes effect for incoming packets with one label only. If you include the `required-depth 2` statement, the `pop-all-labels` statement takes effect for incoming packets with two labels only. If you include the `required-depth [ 1 2 ]` statement, the `pop-all-labels` statement takes effect for incoming packets with one or two labels. A required depth of `[ 1 2 ]` is equivalent to the default behavior of the `pop-all-labels` statement.

When you remove MPLS labels from incoming packets, note the following:

- The `pop-all-labels` statement has no effect on IP packets with three or more MPLS labels except for MX Series routers with MPCs.
- When you enable MPLS label removal, you must configure all ports on a PIC with the same label popping mode and required depth.
- You use the `pop-all-labels` statement to enable passive monitoring applications, not active monitoring applications.
- You cannot apply MPLS filters or accounting to the MPLS labels because the labels are removed as soon as the packet arrives on the interface.
- On ATM2 interfaces, you must use a label value greater than 4095 because the lower range of MPLS labels is reserved for label-switched interface (LSI) and virtual private LAN service (VPLS) support. For more information, see the Junos OS VPNs Library for Routing Devices.
- The following ATM encapsulation types are not supported on interfaces with MPLS label removal:
  - `atm-ccc-cell-relay`
  - `atm-ccc-vc-mux`
  - `atm-mlppp-llc`
  - `atm-tcc-snap`
  - `atm-tcc-vc-mux`

- `ether-over-atm-llc`
- `ether-vpls-over-atm-llc`

Example: Enabling IPv4 Passive Flow Monitoring

The following example shows a complete configuration for enabling passive flow monitoring on an Ethernet interface.

In this example, the Gigabit Ethernet interface can accept all Ethernet packets. It strips VLAN tags (if there are any) and up to two MPLS labels blindly, and passes IPv4 packets to the monitoring interface. With this configuration, it can monitor IPv4, VLAN+IPv4, VLAN+MPLS+IPv4, and VLAN+MPLS+MPLS +IPv4 labeled packets.

The Fast Ethernet interface can accept only packets with VLAN ID 100. All other packets are dropped. With this configuration, it can monitor VLAN (ID=100)+IPv4, VLAN (ID=100)+MPLS+IPv4, and VLAN (ID=100)+MPLS+MPLS+IPv4 labeled packets.

```
[edit firewall]
family inet {
    filter input-monitoring-filter {
        term def {
            then {
                count counter;
                accept;
            }
        }
    }
}
[edit interfaces]
ge-0/0/0 {
    passive-monitor-mode;
    gigether-options {
        mpls {
            pop-all-labels;
        }
    }
    unit 0 {
        family inet {
            filter {
                input input-monitoring-filter;
            }
        }
```

```
    }
}
fe-0/1/0 {
    passive-monitor-mode;
    vlan-tagging;
    fastether-options {
        mpls {
            pop-all-labels required-depth [ 1 2 ];
        }
    }
    unit 0 {
        vlan-id 100;
        family inet {
            filter {
                input input-monitoring-filter;
            }
        }
    }
}
mo-1/0/0 {
    unit 0 {
        family inet {
            receive-options-packets;
            receive-ttl-exceeded;
        }
    }
    unit 1 {
        family inet;
    }
}
[edit forwarding-options]
monitoring mon1 {
    family inet {
        output {
            export-format cflowd-version-5;
            cflowd 192.0.2.2 port 2055;
            interface mo-1/0/0.0 {
                source-address 192.0.2.1;
            }
        }
    }
}
[edit routing-instances]
```

```
monitoring-vrf {
    instance-type vrf;
    interface ge-0/0/0.0;
    interface fe-0/1/0.0;
    interface mo-1/0/0.1;
    route-distinguisher 68:1;
    vrf-import monitoring-vrf-import;
    vrf-export monitoring-vrf-export;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop mo-1/0/0.1;
        }
    }
}
[edit policy-options]
policy-statement monitoring-vrf-import {
    then {
        reject;
    }
}
policy-statement monitoring-vrf-export {
    then {
        reject;
    }
}
```

Example: Enabling IPv6 Passive Flow Monitoring

The following example shows a complete configuration for enabling IPv6 passive flow monitoring on an Ethernet interface.

In this example, the Gigabit Ethernet interface can accept all Ethernet packets. It strips VLAN tags (if there are any) and up to two MPLS labels blindly, and passes IPv6 packets to the monitoring interface. With this configuration, the Gigabit Ethernet interface can monitor IPv6, VLAN+IPv6, VLAN+MPLS +IPv6, and VLAN+MPLS+MPLS+IPv6 labeled packets.

The vlan-tagged Gigabit Ethernet interface can accept only packets with VLAN ID 100. All other packets are dropped. With this configuration, it can monitor VLAN (ID=100)+IPv6, VLAN (ID=100)+MPLS+IPv6, and VLAN (ID=100)+MPLS+MPLS+IPv6 labeled packets.

```
[edit interfaces]
xe-0/1/0 {
```

```
    passive-monitor-mode;
    unit 0 {
        family inet6 {
            filter {
                input port-mirror6;
            }
            address 2001:db8::1/128;
        }
    }
}
    xe-0/1/2 {
        passive-monitor-mode;
        vlan-tagging;
        unit 0 {
            vlan-id 100;
            family inet6 {
                filter {
                    input port-mirror6;
                }
            }
        }
    }
    xe-0/1/1 {
        unit 0 {
            family inet6 {
                address 2001:db8::1/128;
            }
        }

    }
[edit firewall]
family inet6 {
    filter port-mirror6 {
        term term2 {
            then {
                count count_pm;
                port-mirror;
                accept;
            }
        }
    }
}
[edit forwarding options]
```

```
port-mirroring {
    input {
        rate 1;
    }
    family inet6 {
        output {
            interface xe-0/1/1.0 {
                next-hop 2001:db8::3;
            }
            no-filter-check;
        }
    }
}
```

RELATED DOCUMENTATION

Passive Flow Monitoring Overview | 150

Configuring Passive Flow Monitoring

Juniper Networks PICs and MX Series routers support passive flow monitoring. The PICs receive passively monitored network traffic from an input interface (Gigabit Ethernet or 10-Gigabit Ethernet), convert the received packets into flow records, and export them to a flow server for further analysis.

The key configuration hierarchy statement for passive flow monitoring is the `monitoring` statement found at the `[edit forwarding-options]` hierarchy level. At minimum, you must configure a VRF routing instance to direct the traffic to a monitoring services interface for flow processing.

However, there are several options you can use that add complexity to passive flow monitoring. For example, you can configure the router to direct traffic into a routing instance and deliver the traffic into a monitoring group. You can also use port mirroring and filter-based forwarding to copy and redirect traffic. Optionally, you can configure the monitoring station to encrypt flow output before it is sent to a flow server for processing, to send flow records to a flow collector, or to process on-demand monitoring requests with dynamic flow capture.

RELATED DOCUMENTATION

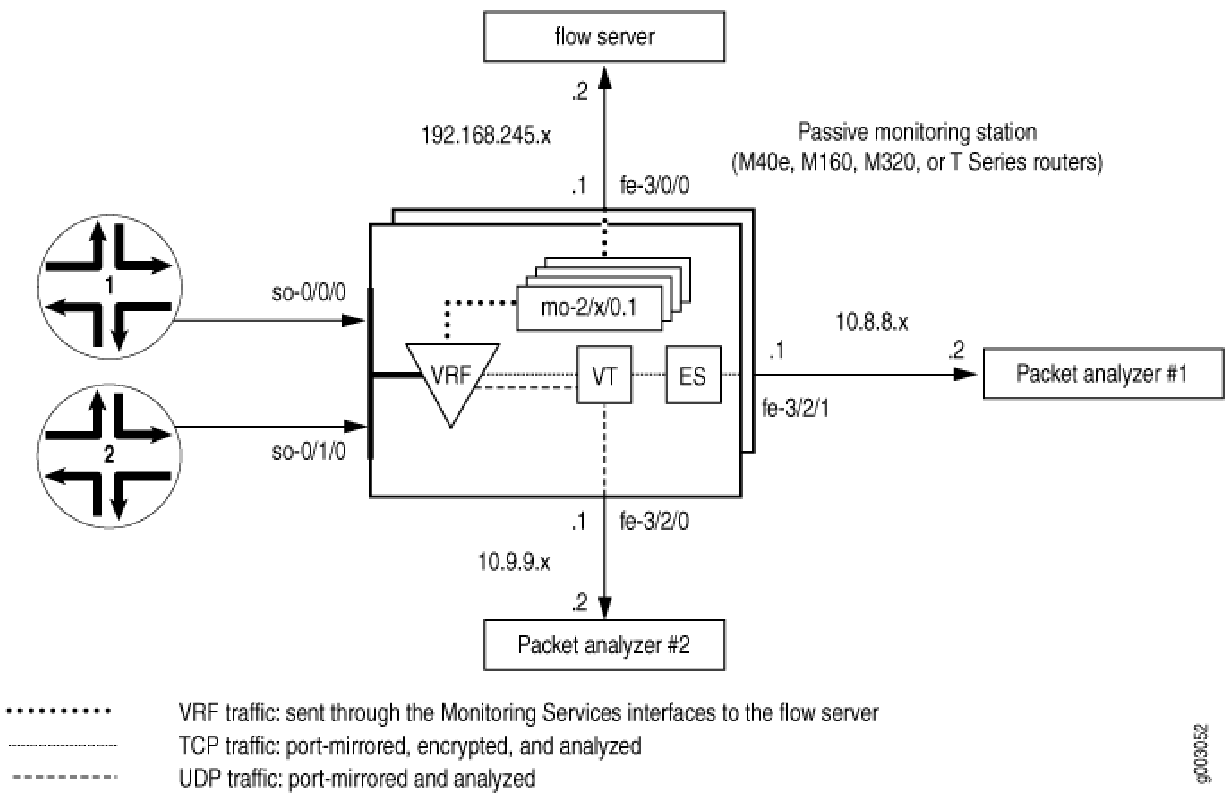
Using an MX Series Router Flow Collector Interface to Process and Export Multiple Flow Records | 197

Passive Flow Monitoring Router and Software Considerations for MX Series Routers | 153

Example: Passive Flow Monitoring Configuration on MX Series Routers

IN THIS SECTION

Figure 22: Passive Flow Monitoring—Topology Diagram



In Figure 22 on page 166, traffic enters the monitoring station through interfaces **so-0/0/0** and **so-0/1/0**. After the firewall filter accepts the traffic to be monitored, the packets enter a VRF instance.

The original packets travel within the VRF instance to the Monitoring Services PIC for flow processing. The final flow packets are sent from the monitoring services interfaces out the **fe-3/0/0** interface to a flow server.

A copy of the accepted traffic is port-mirrored to the Tunnel PIC. As the copied packets enter the tunnel interface, a second firewall filter separates TCP and UDP packets and places them into two filter-based forwarding instances. The UDP instance directs the UDP packets to a packet analyzer attached to **fe-3/2/0**. The TCP instance sends the TCP packets to the ES PIC for encryption and the ES PIC sends the packets to a second packet analyzer connected to **fe-3/2/1**.

Your first step is to define a firewall filter to select packets for monitoring. All filtered traffic must be accepted, and the `port-mirror` statement at the `[edit firewall family inet filter` *`filter-name`* `term` *`term-name`* `then]` hierarchy level facilitates port mirroring.

Next, configure the input SONET/SDH interfaces and apply the firewall filter that you just defined. The `passive-monitor-mode` statement disables SONET keepalives on the SONET/SDH interfaces and enables passive flow monitoring.

Configure all other interfaces that you will use with the monitoring application, including the monitoring services interfaces, the export interfaces, the tunnel interface, and the ES interface. Once the interfaces are in place, configure a VRF instance and monitoring group to direct the original packets from the input interfaces to the monitoring services interfaces for processing. The resulting flow description packets exit **fe-3/0/0** to reach the flow server.

Next, configure statements to port-mirror the monitored traffic to a tunnel interface. Design a firewall filter that selects some of this copied traffic for further analysis and some of the traffic for discarding. In this case, isolate TCP and UDP traffic and direct these two flows into separate filter-based forwarding routing instances. Remember to apply the filter to the tunnel interface to enable the separation of TCP traffic from UDP traffic. Also, import the interface routes into the forwarding instances with a routing table group.

In the filter-based forwarding instances, define static route next hops. The next hop for the TCP instance is the ES interface and the next hop for the UDP instance is the packet analyzer connected to **fe-3/2/0**. Finally, configure IPSec so that the next hop for the TCP traffic is the second packet analyzer attached to **fe-3/2/1**.

```
[edit]
interfaces {
                   so-0/0/0 { # Traffic enters the router on this interface.
        description " input interface";
        encapsulation ppp;
        unit 0 {
                             passive-monitor-mode; # Disables SONET keepalives.
            family inet {
                filter {
                                         input input-monitoring-filter; # The firewall filter is
applied here.
                }
```

```
            }
        }
    }
                so-0/1/0 { # Traffic enters the router on this interface.
        description " input interface";
        encapsulation ppp;
        unit 0 {
                          passive-monitor-mode;  # Disables SONET keepalives.
            family inet {
                filter {
                                      input input-monitoring-filter;  # The firewall filter
is applied here.
                }
            }
        }
    }
                es-3/1/0 {  # This is where the TCP traffic enters the ES PIC.
        unit 0 {
            tunnel {
                source 10.8.8.1;
                destination 10.8.8.2;
            }
            family inet {
                ipsec-sa sa-esp;
                address 192.0.2.1/32 {
                    destination 192.0.2.2;
                }
            }
        }
    }
                fe-3/0/0 { # Flow records exit here and travel to the flow server.
        description " export interface to the flow server";
        unit 0 {
            family inet;
            address 192.168.245.1/30;
        }
    }
                fe-3/2/0 { # This export interface for UDP traffic leads to a packet analyzer.
        description " export interface to the packet analyzer";
        unit 0 {
            family inet {
                address 10.9.9.1/30;
            }
```

```
        }
    }
                fe-3/2/1 {  # This IPSec tunnel source exports TCP traffic to a packet
analyzer.
        unit 0 {
            family inet {
                address 10.8.8.1/30;
            }
        }
    }
                mo-4/0/0 { # This marks the beginning of the monitoring services interfaces.
        unit 0 { # Unit 0 is part of the inet.0 routing table and generates flow records.
            family inet;
        }
                      unit 1 { # Unit 1 receives monitored traffic and is part of the VRF
instance.
            family inet;
        }
    }
                mo-4/1/0 {
        unit 0 { # Unit 0 is part of the inet.0 routing table and generates flow records.
            family inet;
        }
                      unit 1 { # Unit 1 receives monitored traffic and is part of the VRF
instance.
            family inet;
        }
    }
                mo-4/2/0 {
        unit 0 { # Unit 0 is part of the inet.0 routing table and generates flow records.
            family inet;
        }
                      unit 1 { # Unit 1 receives monitored traffic and is part of the VRF
instance.
            family inet;
        }
    }
                mo-4/3/0 {
        unit 0 { # Unit 0 is part of the inet.0 routing table and generates flow records.
            family inet;
        }
                      unit 1 { # Unit 1 receives monitored traffic and is part of the VRF
instance.
```

```
            family inet;
        }
    }
                   vt-0/2/0 {  # The tunnel services interface receives the port-mirrored
traffic.
        unit 0 {
            family inet {
                filter {
                                         input tunnel-interface-filter;  # The filter splits
traffic into TCP and UDP
                }
            }
        }
    }
}
forwarding-options {
                   monitoring group1 {  # Monitored traffic is processed by the monitoring
services
        family inet { # interfaces and flow records are sent to the flow server.
                               output {
                export-format cflowd-version-5;
                flow-active-timeout 60;
                flow-inactive-timeout 30;
                                   flow-server 192.168.245.2 port 2055; # IP address and port
for server.
                                     interface mo-4/0/0.1 {  # Use monitoring services
interfaces for output.
                    engine-id 1; # engine and interface-index statements are optional.
                    engine-type 1;
                    input-interface-index 44;
                    output-interface-index 54;
                                         source-address 192.168.245.1; # This is the IP address
of fe-3/0/0.
                }
                interface mo-4/1/0.1 {
                    engine-id 2; # engine and interface-index statements are optional.
                    engine-type 1;
                    input-interface-index 45;
                    output-interface-index 55;
                    source-address 192.168.245.1; # This is the IP address of fe-3/0/0.
                }
                interface mo-4/2/0.1 {
                    engine-id 3; # engine and interface-index statements are optional.
```

```
                    engine-type 1;
                    input-interface-index 46;
                    output-interface-index 56;
                    source-address 192.168.245.1; # This is the IP address of fe-3/0/0.
                }
                interface mo-4/3/0.1 {
                    engine-id 4; # engine and interface-index statements are optional.
                    engine-type 1;
                    input-interface-index 47;
                    output-interface-index 57;
                                        source-address 192.168.245.1; # This is the IP address
of fe-3/0/0.
                }
            }
        }
    }
                   port-mirroring {  # Copies the traffic and sends it to the Tunnel Services
PIC.
        family inet {
            input {
                rate 1;
                run-length 1;
            }
            output {
                                  interface vt-0/2/0.0;
                no-filter-check;
            }
        }
    }
}
             routing-options { # This installs the interface routes into the forwarding
instances.
                   interface-routes {


         rib-group inet bc-vrf;
    }
    rib-groups {


         bc-vrf {
```

```
         import-rib [inet.0 tcp-routing-table.inet.0 udp-routing-table.inet.0];
        }
    }
    forwarding-table {
        export pplb; # Applies per-packet load balancing to the forwarding table.
    }
}
policy-options {
    policy-statement monitoring-vrf-import {
        then reject;
    }
    policy-statement monitoring-vrf-export {
        then reject;
    }
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
             security { # This sets IPSec options for the ES PIC.
    ipsec {
        proposal esp-sha1-3des {
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 180;
        }
        policy esp-group2 {
            perfect-forward-secrecy {
                keys group2;
            }
            proposals esp-sha1-3des;
        }
        security-association sa-esp {
            mode tunnel;
            dynamic {
                ipsec-policy esp-group2;
            }
        }
    }
    ike {
        proposal ike-esp {
```

```
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 180;
        }
        policy 10.8.8.2 {
            mode aggressive;
            proposals ike-esp;
            pre-shared-key ascii-text "$ABC123";
        }
    }
}
firewall {
    family inet {
        filter input-monitoring-filter { # This filter selects traffic to send into the VRF
            term 1 { # instance and prepares the traffic for port mirroring.
                from {
                    destination-address {
                        10.7.0.0/16;
                    }
                }
                then {
                    port-mirror;
                    accept;
                }
            }
            term 2 {
                from {
                    destination-address {
                        10.6.0.0/16;
                    }
                }
                then accept;
            }
        }
                         filter tunnel-interface-filter {  # This filter breaks the port-
mirrored traffic into two
            term tcp { # filter-based forwarding instances: TCP packets and UDP packets.
                from {
                    protocol tcp;
                }
                then { # This counts TCP packets and sends them into a TCP instance.
```

```
                    count tcp;
                    routing-instance tcp-routing-table;
                }
            }
            term udp {
                from {
                    protocol udp;
                }
                then { # This counts UDP packets and sends them into a UDP instance.
                    count udp;
                    routing-instance udp-routing-table;
                }
            }
            term rest {
                then {
                    count rest;
                    discard;
                }
            }
        }
    }
}
routing-instances {
                   monitoring-vrf { # This is the VRF instance where you send the traffic. It
contains
        instance-type vrf; # the input interface and the monitoring services interfaces.
                       interface so-0/0/0.0; # Traffic enters the router on these input
interfaces.
        interface so-0/1/0.0;


         interface mo-4/0/0.1;



         interface mo-4/1/0.1; # These are output interfaces (use them as
        interface mo-4/2/0.1; # output interfaces in your monitoring group).
        interface mo-4/3/0.1;
        route-distinguisher 69:1;
        vrf-import monitoring-vrf-import;
        vrf-export monitoring-vrf-export;
        routing-options { # Sends traffic to a group of monitoring services interfaces.
            static {
```

```
         route 0.0.0.0/0 next-hop [mo-4/0/0.1 mo-4/1/0.1
mo-4/2/0.1 mo-4/3/0.1];
            }
        }
    }
                   tcp-routing-table {  # This is the filter-based forwarding instance for TCP
traffic.
        instance-type forwarding;
                         routing-options {  # The next hop is the ES PIC.
            static {


         route 0.0.0.0/0 next-hop es-3/1/0.0;
            }
        }
    }
    udp-routing-table { # This is the filter-based forwarding instance for UDP traffic.
        instance-type forwarding;
        routing-options { # The next hop is the second packet analyzer.
            static {


         route 0.0.0.0/0 next-hop 10.9.1.2;
            }
        }
    }
}
```

Verifying Your Work

To verify that your configuration is correct, use the following commands on the monitoring station that is configured for passive flow monitoring:

- `show route 0/0`
- `show passive-monitoring error`
- `show passive-monitoring flow`
- `show passive-monitoring memory`
- `show passive-monitoring status`

- `show passive-monitoring usage`

To clear statistics for the `show passive-monitoring error` and `show passive-monitoring flow` commands, issue the `clear passive-monitoring (all | interface-name)` command.

You can also view passive flow monitoring status with the Simple Network Management Protocol (SNMP). The following Management Information Base (MIB) tables are supported:

- **jnxPMonErrorTable**—Corresponds to the `show passive-monitoring error` command.
- **jnxPMonFlowTable**—Corresponds to the `show passive-monitoring flow` command.
- **jnxPMonMemoryTable**—Corresponds to the `show passive-monitoring memory` command.

The following section shows the output of the `show` commands used with the configuration example:

```
user@host> show route 0/0
 <skip inet.0>
```

We are only concerned with the routing-instance route.

```
bc-vrf.inet.0:  1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
bc-vrf.inet.0:+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 5d 17:34:57
                      via mo-4/0/0.1
                    > via mo-4/1/0.1
                      via mo-4/2/0.1
                      via mo-4/3/0.1
tcp-rt.inet.0: 13 destinations, 13 routes (12 active, 0 holddown, 1
hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 19:24:39
                    > via es-3/1/0.0
  : <other interface routes>
udp-rt.inet.0: 13 destinations, 13 routes (12 active, 0 holddown, 1
hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[Static/5] 19:24:39
                    > to 10.9.1.2 via fe-3/2/0.0
  : <other interface routes>
```

NOTE: For all `show passive-monitoring` commands, the output obtained when using a wildcard (such as *) or the `all` option is based on the configured interfaces listed at the `[edit forwarding-options monitoring *group-name*]` hierarchy level. In the output from the configuration example, you see information only for the configured interfaces `mo-4/0/0`, `mo-4/1/0`, `mo-4/2/0`, and `mo-4/3/0`.

Many of the statements you can configure in a monitoring group, such as `engine-id` and `engine-type`, are visible in the output of the `show passive-monitoring` commands.

Table 26: Output Fields for the show passive-monitoring error Command

Field	Explanation
Packets dropped (no memory)	Number of packets dropped because of memory.
Packets dropped (not IP)	Number of non-IP packets dropped.
Packets dropped (not IPv4)	Number of packets dropped because they failed the IPv4 check.
Packets dropped (header too small)	Number of packets dropped because the packet length or IP header length was too small.
Memory allocation failures	Number of flow record memory allocation failures. A small number reflects failures to replenish the free list. A large number indicates the monitoring station is almost out of memory space.
Memory free failures	Number of flow record memory frees.
Memory free list failures	Number of flow records received from free list that failed. Memory is nearly exhausted or too many new flows greater than 128K are being created in one second.

Table 26: Output Fields for the show passive-monitoring error Command *(Continued)*

Field	Explanation
Memory warning	The flows have exceeded 1 million packets per second (Mpps) on a Monitoring Services PIC or 2 Mpps on a Monitoring Services II PIC. The response can be **Yes** or **No**.
Memory overload	The memory has been overloaded. The response is **Yes** or **No**.
PPS overload	In packets per second, whether the PIC is receiving more traffic than the configured threshold. The response can be **Yes** or **No**.
BPS overload	In bytes per second, whether the PIC is receiving more traffic than the configured threshold. The response can be **Yes** or **No**.

```
user@host> show passive-monitoring error all
Passive monitoring interface: mo-4/0/0, Local interface index: 44
  Error information
    Packets dropped (no memory): 0, Packets dropped (not IP): 0
    Packets dropped (not IPv4): 0, Packets dropped (header too small): 0
    Memory allocation failures: 0, Memory free failures: 0
    Memory free list failures: 0
    Memory warning: No, Memory overload: No, PPS overload: No, BPS overload: No

Passive monitoring interface: mo-4/1/0, Local interface index: 45
  Error information
    Packets dropped (no memory): 0, Packets dropped (not IP): 0
    Packets dropped (not IPv4): 0, Packets dropped (header too small): 0
    Memory allocation failures: 0, Memory free failures: 0
    Memory free list failures: 0
    Memory warning: No, Memory overload: No, PPS overload: No, BPS overload: No

Passive monitoring interface: mo-4/2/0, Local interface index: 46
  Error information
    Packets dropped (no memory): 0, Packets dropped (not IP): 0
```

```
    Packets dropped (not IPv4): 0, Packets dropped (header too small): 0
    Memory allocation failures: 0, Memory free failures: 0
    Memory free list failures: 0
    Memory warning: No, Memory overload: No, PPS overload: No, BPS overload: No

Passive monitoring interface: mo-4/3/0, Local interface index: 47
  Error information
    Packets dropped (no memory): 0, Packets dropped (not IP): 0
    Packets dropped (not IPv4): 0, Packets dropped (header too small): 0
    Memory allocation failures: 0, Memory free failures: 0
    Memory free list failures: 0
    Memory warning: No, Memory overload: No, PPS overload: No, BPS overload: No
```

Table 27: Output Fields for the show passive-monitoring flow Command

Field	Explanation
Flow packets	Number of packets received by an operational PIC.
Flow bytes	Number of bytes received by an operational PIC.
Flow packets 10-second rate	Number of packets per second handled by the PIC and displayed as a 10-second average.
Flow bytes 10-second rate	Number of bytes per second handled by the PIC and displayed as a 10-second average.
Active flows	Number of currently active flows tracked by the PIC.
Total flows	Total number of flows received by an operational PIC.
Flows exported	Total number of flows exported by an operational PIC.
Flows packets exported	Total number of flow packets exported by an operational PIC.
Flows inactive timed out	Total number of flows that are exported because of inactivity.

Table 27: Output Fields for the show passive-monitoring flow Command *(Continued)*

Field	Explanation
Flows active timed out	Total number of long-lived flows that are exported because of an active timeout.

```
user@host> show passive-monitoring flow all
Passive monitoring interface: mo-4/0/0, Local interface index: 44
  Flow information
    Flow packets: 6533434, Flow bytes: 653343400
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 1599
    Flows exported: 1599, Flows packets exported: 55
    Flows inactive timed out: 1599, Flows active timed out: 0

Passive monitoring interface: mo-4/1/0, Local interface index: 45
  Flow information
    Flow packets: 6537780, Flow bytes: 653778000
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 1601
    Flows exported: 1601, Flows packets exported: 55
    Flows inactive timed out: 1601, Flows active timed out: 0

Passive monitoring interface: mo-4/2/0, Local interface index: 46
  Flow information
    Flow packets: 6529259, Flow bytes: 652925900
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 1599
    Flows exported: 1599, Flows packets exported: 55
    Flows inactive timed out: 1599, Flows active timed out: 0

Passive monitoring interface: mo-4/3/0, Local interface index: 47
  Flow information
    Flow packets: 6560741, Flow bytes: 656074100
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 1598
    Flows exported: 1598, Flows packets exported: 55
    Flows inactive timed out: 1598, Flows active timed out: 0
```

Table 28: Output Fields for the show passive-monitoring memory Command

Field	Explanation
Allocation count	Number of flow records allocated.
Free count	Number of flow records freed.
Maximum allocated	Maximum number of flow records allocated since the monitoring station booted. This number represents the peak number of flow records allocated at a time.
Allocations per second	Flow records allocated per second during the last statistics interval on the PIC.
Frees per second	Flow records freed per second during the last statistics interval on the PIC.
Total memory used	Total amount of memory currently used (in bytes).
Total memory free	Total amount of memory currently free (in bytes).

```
user@host> show passive-monitoring memory all
Passive monitoring interface: mo-4/0/0, Local interface index: 44
  Memory utilization
    Allocation count: 1600, Free count: 1599, Maximum allocated: 1600
    Allocations per second: 3200, Frees per second: 1438
    Total memory used (in bytes): 103579176, Total memory free (in bytes): 163914184

Passive monitoring interface: mo-4/1/0, Local interface index: 45
  Memory utilization
    Allocation count: 1602, Free count: 1601, Maximum allocated: 1602
    Allocations per second: 3204, Frees per second: 1472
    Total memory used (in bytes): 103579176, Total memory free (in bytes): 163914184

Passive monitoring interface: mo-4/2/0, Local interface index: 46
  Memory utilization
    Allocation count: 1600, Free count: 1599, Maximum allocated: 1600
```

```
    Allocations per second: 3200, Frees per second: 1440
    Total memory used (in bytes): 103579176, Total memory free (in bytes): 163914184

Passive monitoring interface: mo-4/3/0, Local interface index: 47
  Memory utilization
    Allocation count: 1599, Free count: 1598, Maximum allocated: 1599
    Allocations per second: 3198, Frees per second: 1468
    Total memory used (in bytes): 103579176, Total memory free (in bytes): 163914184
```

Table 29: Output Fields for the show passive-monitoring status Command

Field	Explanation
Interface state	Indicates whether the interface is monitoring (operating properly), disabled (administratively disabled), or not monitoring (not configured).
Group index	Integer that represents the monitoring group of which the PIC is a member. (This does not indicate the number of monitoring groups.)
Export interval	Configured export interval for flow records, in seconds.
Export format	Configured export format (only v5 is currently supported).
Protocol	Protocol the PIC is configured to monitor (only IPv4 is currently supported).
Engine type	Configured engine type that is inserted in output flow packets.
Engine ID	Configured engine ID that is inserted in output flow packets.
Route record count	Number of routes recorded.
IFL to SNMP index count	Number of logical interfaces mapped to an SNMP index.
AS count	Number of AS boundaries that the flow has crossed.
Time set	Indicates whether the time stamp is in place.

Table 29: Output Fields for the show passive-monitoring status Command *(Continued)*

Field	Explanation
Configuration set	Indicates whether the monitoring configuration is set.
Route record set	Indicates whether routes are being recorded.
IFL SNMP map set	Indicates whether logical interfaces are being mapped to an SNMP index.

```
user@host> show passive-monitoring status all
Passive monitoring interface: mo-4/0/0, Local interface index: 44
  Interface state: Monitoring
  Group index: 0
  Export interval: 15 secs, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 1
  Route record count: 13, IFL to SNMP index count: 30, AS count: 1
  Time set: Yes, Configuration set: Yes
  Route record set: Yes, IFL SNMP map set: Yes

Passive monitoring interface: mo-4/1/0, Local interface index: 45
  Interface state: Monitoring
  Group index: 0
  Export interval: 15 secs, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 2
  Route record count: 13, IFL to SNMP index count: 30, AS count: 1
  Time set: Yes, Configuration set: Yes
  Route record set: Yes, IFL SNMP map set: Yes

Passive monitoring interface: mo-4/2/0, Local interface index: 46
  Interface state: Monitoring
  Group index: 0
  Export interval: 15 secs, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 3
  Route record count: 13, IFL to SNMP index count: 30, AS count: 1
  Time set: Yes, Configuration set: Yes
  Route record set: Yes, IFL SNMP map set: Yes

Passive monitoring interface: mo-4/3/0, Local interface index: 47
```

```
  Interface state: Monitoring
  Group index: 0
  Export interval: 15 secs, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 4
  Route record count: 13, IFL to SNMP index count: 30, AS count: 1
  Time set: Yes, Configuration set: Yes
  Route record set: Yes, IFL SNMP map set: Yes
```

Table 30: Output Fields for the show passive-monitoring usage Command

Field	Explanation
Uptime	Time, in milliseconds, that the PIC has been operational.
Interrupt time	Cumulative time that the PIC spent in processing packets since the last PIC reset.
Load (5 second)	CPU load on the PIC averaged over 5 seconds. The number is a percentage obtained by dividing the time spent on active tasks by the total elapsed time.
Load (1 minute)	CPU load on the PIC averaged over 1 minute. The number is a percentage obtained by dividing the time spent on active tasks by the total elapsed time.

```
user@host> show passive-monitoring usage *
Passive monitoring interface: mo-4/0/0, Local interface index: 44
  CPU utilization
    Uptime: 653155 milliseconds, Interrupt time: 40213754 microseconds
    Load (5 second): 20%, Load (1 minute): 17%

Passive monitoring interface: mo-4/1/0, Local interface index: 45
  CPU utilization
    Uptime: 652292 milliseconds, Interrupt time: 40223178 microseconds
    Load (5 second): 22%, Load (1 minute): 15%

Passive monitoring interface: mo-4/2/0, Local interface index: 46
  CPU utilization
```

```
    Uptime: 649491 milliseconds, Interrupt time: 40173645 microseconds
    Load (5 second): 22%, Load (1 minute): 10098862%

Passive monitoring interface: mo-4/3/0, Local interface index: 47
  CPU utilization
    Uptime: 657328 milliseconds, Interrupt time: 40368704 microseconds
    Load (5 second): 1%, Load (1 minute): 15%
```

Configuring a Routing Table Group on an M, MX or T Series Router to Add Interface Routes into the Forwarding Instance

Next, import the interface routes into the forwarding instance. This step is necessary because the next hops specified in the forwarding instances must be installed in the forwarding instances themselves. To configure, include the `import-rib` statement at the `[edit routing-options rib-groups` *`group-name`*`]` hierarchy level. The `export` statement at the `[edit routing-options forwarding-table]` hierarchy level and the **pplb** policy enable load balancing.

```
[edit]
routing-options {
    interface-routes {
        rib-group inet bc-vrf;
    }
    rib-groups {
        bc-vrf {
            import-rib [inet.0 tcp-routing-table.inet.0 udp-routing-table.inet.0];
        }
    }
    forwarding-table {
        export pplb;
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

Using IPSec and an ES PIC on an M, MX or T Series Router to Send Encrypted Traffic to a Packet Analyzer

You can send some or all of the traffic securely to the packet analyzer using IPSec (a suite of related protocols for cryptographically securing communications at the IP Packet Layer) and an Encryption Services (ES) PIC. In this case, the TCP traffic is encrypted, sent over an IPSec tunnel, and received by the packet analyzer. For more information on configuring IPSec on the ES PIC, see the *IPsec User Guide* or the *Junos System Basics Configuration Guide*.

```
[edit]
interfaces {
    es-3/1/0 {
        unit 0 {
            tunnel {
                source 10.8.8.1;
                destination 10.8.8.2;
            }
            family inet {
                ipsec-sa sa-esp;
                address 192.0.2.1/32 {
                    destination 192.0.2.2;
                }
            }
        }
    }
    fe-3/2/1 {
        unit 0 {
            family inet {
                address 10.8.8.1/30;
            }
        }
    }
}
security {
    ipsec {
        proposal esp-sha1-3des {
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 180;
        }
```

```
        policy esp-group2 {
            perfect-forward-secrecy {
                keys group2;
            }
            proposals esp-sha1-3des;
        }
        security-association sa-esp {
            mode tunnel;
            dynamic {
                ipsec-policy esp-group2;
            }
        }
    }
    ike {
        proposal ike-esp {
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 180;
        }
        policy 10.8.8.2 {
            mode aggressive;
            proposals ike-esp;
            pre-shared-key ascii-text "$ABC123";
        }
    }
}
```

Applying a Firewall Filter Output Interface on an M, MX or T Series Router to Port-mirror Traffic to PICs or Flow Collection Services

On output interfaces, you can apply a firewall filter that leads to a filter-based forwarding routing instance. This is useful if you want to port-mirror traffic to multiple Monitoring Services PICs or flow collection services interfaces. To configure, include the `output` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet filter]` hierarchy level.

```
[edit]
interfaces
```

```
fe-3/1/0 {
    description “export interface to flow collection services interfaces”;
    unit 0 {
        family inet;
        address ip-address;
        filter {
            output output-filter-name;
        }
    }
}
```

Monitoring Traffic on a Router with a VRF Instance and a Monitoring Group

The first way you can implement passive flow monitoring is to direct traffic into a VRF routing instance and use a monitoring group to export this traffic to a flow server for analysis. Complete the following tasks:

- "Specifying a Firewall Filter on an M, MX or T Series Router to Select Traffic to Monitor" on page 189
- "Configuring Input Interfaces, Monitoring Services Interfaces and Export Interfaces on M, MX or T Series Routers" on page 190
- "Establishing a VRF Instance on an M, MX or T Series Router for Monitored Traffic" on page 192
- "Configuring a Monitoring Group on an M, MX or T Series Router to Send Traffic to the Flow Server" on page 193
- "Configuring Policy Options on M, MX or T Series Routers" on page 195
- "Stripping MPLS Labels on ATM, Ethernet-Based and SONET/SDH Router Interfaces" on page 196

Specifying a Firewall Filter on an M, MX or T Series Router to Select Traffic to Monitor

When you define a firewall filter, you select the initial traffic to be monitored. To configure a firewall filter, include the `filter` statement at the `[edit firewall family inet]` hierarchy level. All filtered traffic to be monitored must be accepted.

```
[edit]
firewall {
    family inet {
        filter input-monitoring-filter {
            term 1 {
                from {
                    destination-address {
                        10.7.0.0/16;
                    }
                }
                then {
                    count counter1;
                    accept;
                }
            }
            term 2 {
                from {
                    destination-address {
                        10.6.0.0/16;
                    }
                }
                then {
                    count counter2;
                    accept;
                }
            }
        }
    }
}
```

Configuring Input Interfaces, Monitoring Services Interfaces, and Export Interfaces on MX Series Routers

After creating the input filter, you need to configure the interfaces where traffic will enter the router.

Ethernet-based interfaces support both per-port passive monitoring and per-VLAN passive monitoring. For Gigabit Ethernet interfaces, include the `passive-monitor-mode` statement at the `[edit interfaces ge-`*`fpc/pic/port`*`]` hierarchy level. On Ethernet-based interfaces, passive monitor mode disables the Routing Engine from receiving packets and prevents the routing table from transmitting packets. You can verify this by the presence of the `No-receive` and `No-transmit` interface flags in the output of the `show interfaces ge-`*`fpc/pic/port`* command.

NOTE: The following restrictions apply to passive flow monitoring on Ethernet-based interfaces:

- No special encapsulation types are allowed, so you must configure Ethernet encapsulations only.
- When you configure the `passive-monitor-mode` statement, destination MAC address filters applied to incoming interfaces are disabled by default.
- The `flow-control` statement at the `[edit interfaces ge-`*`fpc/pic/port`*` gigether-options]` hierarchy level does not work when passive flow monitoring is enabled.

In addition to passive monitor mode, apply the previously defined firewall filter to the interface with the `filter` statement at the `[edit interfaces `*`interface-name-fpc/pic/port`*` unit `*`unit-number`*` family inet]` hierarchy level:

```
[edit]
interfaces {
    ge-2/0/0 {
        description “Gigabit Ethernet input interface”;
        passive-monitor-mode;
        unit 0 {
            family inet {
                filter {
                    input input-monitoring-filter;
                }
            }
        }
```

```
    }
}
```

Configure the interfaces on the Monitoring Services PIC or Monitoring Services II PIC with the `family inet` statement at the `[edit interfaces mo-`*`fpc/pic/port`*` unit `*`unit-number`*`]` hierarchy level. The statement allows the interfaces to process IPv4 traffic received from the input interfaces.

When you use VRF instances, you need to configure two logical interfaces. The first (`unit 0`) is part of the inet.0 routing table and sources the flow packets. The second (`unit 1`) is configured as part of the VRF instance so the monitoring services interface can serve as a valid next hop for packets received in the instance.

You can also capture options packets and time-to-live (TTL) exceeded information when the monitoring services interface processes flow records. To configure, include the `receive-options-packets` and `receive-ttl-exceeded` statements at the `[edit interfaces mo-`*`fpc/pic/port`*` unit `*`unit-number`*` family inet]` hierarchy level:

```
[edit]
interfaces {
    mo-4/0/0 {
        unit 0 {
            family inet {
                receive-options-packets;
                receive-ttl-exceeded;
            }
        }
        unit 1 {
            family inet;
        }
    }
    mo-4/1/0 {
        unit 0 {
            family inet;
        }
        unit 1 {
            family inet;
        }
    }
    mo-4/2/0 {
        unit 0 {
            family inet;
        }
        unit 1 {
            family inet;
```

```
        }
    }
    mo-4/3/0 {
        unit 0 {
            family inet;
        }
        unit 1 {
            family inet;
        }
    }
}
```

You must also configure the export interface where flow packets exit the monitoring station and are sent to the flow server.

On output interfaces, you can apply a firewall filter that leads to a filter-based forwarding routing instance. This is useful if you want to port-mirror traffic to multiple Monitoring Services PICs or flow collection services interfaces. To configure, include the `output` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet filter]` hierarchy level. For more information, see Using Filter-Based Forwarding to Export Monitored Traffic to Multiple Destinations.

```
[edit]
interfaces
ge-3/0/0 {
    description “export interface to flow server”;
    unit 0 {
        family inet;
        address ip-address;
        filter {
            output output-filter-name;
        }
    }
}
```

Establishing a VRF Instance on an M, MX or T Series Router for Monitored Traffic

After the firewall filter and interfaces are ready, create a VPN routing and forwarding (VRF) instance. The filtered traffic enters the VRF instance and is shared only between the input interfaces and the monitoring services output interfaces. In this case, a group of four monitoring services interfaces is used as the next hop.

```
[edit]
routing-instances {
    monitoring-vrf {
        instance-type vrf;
        interface so-0/0/0.0;
        interface so-0/1/0.0;
        interface mo-4/0/0.1;
        interface mo-4/1/0.1;
        interface mo-4/2/0.1;
        route-distinguisher 69:1;
        vrf-import monitoring-vrf-import;
        vrf-export monitoring-vrf-export;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop [mo-4/0/0.1 mo-4/1/0.1 mo-4/2/0.1];
            }
        }
    }
}
```

Configuring a Monitoring Group on an MX Series Router to Send Traffic to the Flow Server

You collect flow records by specifying output interfaces in a monitoring group. In general, the monitoring services interfaces are the output interfaces. The logical unit number on the output interfaces when used in conjunction with a VRF instance must be 1. To configure, include the `output` statement at the `[edit forwarding-options monitoring` *`group-name`* `family inet]` hierarchy level.

As part of the `mo-`*`fpc/pic/port`* statement at the `[edit forwarding-options monitoring` *`group-name`* `family inet output interface]` hierarchy level, you must specify a source address for transmission of flow information. You can use the router ID IP address, the IP address of the input interface, or any local IP address of your choice as the source address. If you provide a different `source-address` statement for each monitoring services output interface, you can track which interface processes a particular flow record.

All other statements at this level (`engine-id`, `engine-type`, `input-interface-index`, and `output-interface-index`) are dynamically generated, but can be configured manually. To reset outgoing interface or incoming interface indexes that were once configured manually, configure the `input-interface-index` or `outgoing-interface-index` statements with a value of 0 at the `[edit forwarding-options monitoring group-name family inet output interface interface-name]` hierarchy level.

To specify the flow server IP address and port number, include the `flow-server ip-address port port-number` statement at the `[edit forwarding-options monitoring group-name family inet output]` hierarchy level. You can specify up to eight flow servers in a monitoring group and the IP address for each server must be unique. Flow records are exported and load-balanced between all active flow servers.

Once you configure the VRF and monitoring group statements, traffic enters the input interfaces, passes to the monitoring services interfaces for processing, and is discarded. The resulting flow description packets exit the monitoring station through the export interface. If you want traffic to travel to destinations other than the monitoring services interfaces, or need to establish additional analysis, see the section Copying and Redirecting Traffic with Port Mirroring and Filter-Based Forwarding.

NOTE: You must complete interface configuration on the Monitoring Services or Monitoring Services II PIC before an interface can be added into a monitoring group. For more information, see "Configuring Input Interfaces, Monitoring Services Interfaces and Export Interfaces on M, MX or T Series Routers" on page 190.

```
[edit]
forwarding-options {
    monitoring group1 {
        family inet {
            output {
                export-format cflowd-version-5;
                flow-active-timeout 60;
                flow-inactive-timeout 30;
                flow-server 192.168.245.1 port 2055;
                flow-server 192.168.245.2 port 2055;
                interface mo-4/0/0.1 {
                    engine-id 1;
                    engine-type 1;
                    input-interface-index 44;
                    output-interface-index 54;
                    source-address 192.168.245.1;
                }
                interface mo-4/1/0.1 {
                    engine-id 2;
                    engine-type 1;
                    input-interface-index 45;
                    output-interface-index 55;
```

```
                    source-address 192.168.245.1;
                }
                interface mo-4/2/0.1 {
                    engine-id 3;
                    engine-type 1;
                    input-interface-index 46;
                    output-interface-index 56;
                    source-address 192.168.245.1;
                }
            }
        }
    }
}
```

Configuring Policy Options on M, MX or T Series Routers

When you use a group of next hops in your monitoring group, you can load-balance traffic and distribute it to the export interfaces if you configure policy options. To configure, include the `load-balance per-packet` statement at the `[edit policy-options policy-statement` *`policy-name`* `then]` hierarchy level. You can also reject import and export of VRF routes by including the `reject` statement at the `[edit policy-options policy-statement` *`policy-name`* `then]` hierarchy level.

```
[edit]
routing-options {
    forwarding-table {
        export pplb;
    }
}
policy-options {
    policy-statement monitoring-vrf-import {
        then {
            reject;
        }
    }
    policy-statement monitoring-vrf-export {
        then {
            reject;
        }
    }
```

```
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
```

Stripping MPLS Labels on Ethernet-Based Router Interfaces

Because flow monitoring can be performed only on IPv4 packets, any packets containing MPLS labels must have the labels removed before monitoring can occur. To remove MPLS labels from packets as they enter an Ethernet-based interface, include the `pop-all-labels` statement at the `[edit interfaces ge-`*`fpc/pic/port`*` gigether-options mpls]` hierarchy level. If you use static MPLS labels, we recommend you assign label values from `10000` through `99999` to avoid using the label ranges reserved by the Junos OS.

To remove a specified number of labels from selected packets with MPLS labels, include the `required-depth` statement at the `[edit interfaces `*`interface-name-fpc/pic/port`*` gigether-options mpls pop-all-labels]` hierarchy level. A `required-depth` value of `1` removes labels from all packets containing only one MPLS label, a value of `2` removes labels from all packets containing only two MPLS labels, and a value of `[1 2]` removes labels from all packets containing either one or two MPLS labels. The `required-depth` value of `[1 2]` is the default setting. When you configure the `required-depth` statement, you must configure the same value for all ports on the same PIC.

The labels are removed and discarded as soon as they arrive at the interface. As a result, no MPLS filters can be applied to the stripped labels, no statistics are generated for the labels, and you cannot apply an IP filter to the incoming packets. No Tunnel Services PIC is required to perform MPLS label stripping.

```
[edit]
interfaces {
    ge-fpc/pic/port {
        gigether-options {
            mpls {
                pop-all-labels {
                    required-depth [1 2];
                }
            }
        }
    }
}
```

Using an MX Series Router Flow Collector Interface to Process and Export Multiple Flow Records

Basic passive monitoring can sometimes create a large number of flow records. However, you can manage multiple flow records with a flow collector interface. You can create a flow collector interface from a Monitoring Services II PIC. The flow collector interface combines multiple flow records received from a monitoring services interface into a compressed ASCII data file and exports the file to an FTP server.

To convert a Monitoring Services II PIC into a flow collector interface, include the `flow-collector` statement at the `[edit chassis fpc` *`fpc-slot`* `pic` *`pic-slot`* `monitoring-services application]` hierarchy level. To restore the monitoring functions of a Monitoring Services II PIC, include the `monitor` statement at the `[edit chassis fpc` *`fpc-slot`* `pic` *`pic-slot`* `monitoring-services application]` hierarchy level.

After you commit the configuration to convert the PIC between the `monitor` and `flow-collector` service types, you must take the PIC offline and then bring the PIC back online. Rebooting the router does not enable the new service type. You can use the Monitoring Services II PIC for either flow collection or monitoring, but not both types of service simultaneously.

A flow collector interface, designated by the `cp-`*`fpc/pic/port`* interface name, requires three logical interfaces for correct operation. Units 0 and 1 are used respectively as export channels 0 and 1 to send the compressed ASCII data files to an FTP server. You must include a class-of-service (CoS) configuration for these two export channels to provide adequate bandwidth for file transmission. Unit 2 is used as a flow receive channel to receive flow records from a monitoring services interface.

> **NOTE**: Unlike conventional interfaces, IP addresses for flow collector logical interfaces set up a point-to-point connection between the Routing Engine and the flow collector. The `address` statement at the `[edit interfaces cp-`*`fpc/pic/port`* `unit` *`unit-number`* `family inet]` hierarchy level corresponds to the IP address of the Routing Engine. Likewise, the `destination` statement at the `[edit interfaces cp-`*`fpc/pic/port`* `unit` *`unit-number`* `family inet address` *`ip-address`*`]` hierarchy level corresponds to the IP address of the flow collector interface. As a result, you must configure the `destination` statement for Units 0 and 1 (export channels 0 and 1) with **local** addresses that can reach the FTP server. Similarly, configure the `destination` statement for Unit 2 (flow receive channel) with a **local** IP address so it can reach the monitoring services interface that sends flow records.

To activate flow collector services after the Monitoring Services II PIC is converted into a flow collector, include the `flow-collector` statement at the `[edit services]` hierarchy level. You also need to configure several additional components:

- Destination of the FTP server—Determines where the compressed ASCII data files are sent after the flow records are collected and processed. To specify the destination FTP server, include the destinations statement at the [edit services flow-collector] hierarchy level. You can specify up to two FTP server destinations and include the password for each configured server. If two FTP servers are configured, the first server in the configuration is the primary server and the second is a backup server.

- File specifications—Preset data file formats, name formats, and transfer characteristics. Files are sent by FTP to the destination FTP server when the timer expires or when a preset number of records are received, whichever comes first. To set the data file format, include the data-format statement at the [edit services flow-collector file-specification *file-name*] hierarchy level. The default data format is flow-compressed. To set the export timer and file size thresholds, include the transfer statement at the [edit services flow-collector file-specification *file-name*] hierarchy level and specify values for the timeout and record-level options. The default values are 600 seconds for timeout and 500,000 records for record-level.

  To set the filename format, include the name-format statement at the [edit services flow-collector file-specification *file-name*] hierarchy level. Common name format macros that you can use in your configuration are included in Table 31 on page 198.

Table 31: Name Format Macros

Field	Expansion
{*am_pm*}	AM or PM
{*date*}	Expands to the current date, using the {month}, {day}, and {year} macros.
{*day*}	01 to 31
{*day_abbr*}	Sun through Sat
{*day_full*}	Sunday through Saturday
{*generation_number*}	Expands to a unique, sequential number for each new file created.
{*hour_12*}	01 to 12
{*hour_24*}	00 to 23

Table 31: Name Format Macros *(Continued)*

Field	Expansion
{*ifalias*}	Expands to a description string for the logical interface.
{*minute*}	`00` to `59`
{*month*}	`01` to `12`
{*month_abbr*}	`Jan` through `Dec`
{*month_full*}	`January` through `December`
{*num_zone*}	`-2359` to `+2359`
{*second*}	`00` to `60`
{*time*}	Expands to the time the file is created, using the `{hour_24}`, `{minute}`, and `{second}` macros.
{*time_zone*}	Time zone code name of the locale (`gmt`, `pst`, and so on).
{*year*}	`1970`, `2008`, and so on.
{*year_abbr*}	`00` to `99`

- Input interface-to-flow collector interface mappings—Match an input interface with a flow collector interface and apply the preset file specifications to the input interface. To configure the default flow collector and file specifications for all input interfaces, include the `file-specification` and `collector` statements at the `[edit services flow-collector interface-map]` hierarchy level. To override the default settings and apply flow collector and file specifications to a specific input interface, include the `file-specification` and `collector` statements at the `[edit services flow-collector interface-map *interface-name*]` hierarchy level.
- Transfer log settings—Allow you to configure the destination FTP server where log files containing the transfer activity history for a flow collector interface are to be archived, the name for the log file,

and the amount of time the router waits before sending the log file to the FTP server. To configure, include the `archive-sites`, `filename-prefix`, and `maximum-age` statements at the `[edit services flow-collector transfer-log-archive]` hierarchy level. The default value for the `maximum-age` statement is 120 minutes, with a range of 1 to 360 minutes. Also, you can configure up to five FTP archive site servers to receive log files.

- Miscellaneous settings—Allow you to configure values for the IP address of the analyzer, an identifier for the analyzer, the maximum number of times the flow collector interface attempts to send transfer log files to the FTP server, and the amount of time the flow collector interface waits between retry attempts. To configure, include the `analyzer-address`, `analyzer-id`, `retry`, and `retry-delay` statements at the `[edit services flow-collector]` hierarchy level. The range for the `retry` statement is 0 through 10 retry attempts. The default for the `retry-delay` statement is 30 seconds and the range is 0 through 60 seconds.

To specify a flow collector interface as the destination for flow records coming from a Monitoring Services or Monitoring Services II PIC, include the `collector-pic` statement at the `[edit forwarding-options monitoring` *`group-name`* `family inet output flow-export-destination]` hierarchy level. You can select either the flow collector interface or a flow server as the destination for flow records, but you cannot select both destination types simultaneously.

There is also a Juniper Networks enterprise Management Information Base (MIB) for the flow collector interface. The Flow Collector Services MIB allows you to use SNMP to monitor the flow collector interface. The MIB provides statistics on files, records, memory, FTP, and error states of a flow collector interface. It also provides SNMP traps for unavailable destinations, unsuccessful file transfers, flow overloading, and memory overloading. For more information, view the enterprise-specific Juniper Networks MIBs at SNMP MIB Explorer.

In summary, to implement the flow collector service, include statements at the `[edit chassis]`, `[edit interfaces]`, `[edit forwarding-options]`, and `[edit services]` hierarchy levels. The excerpt on the following pages shows the flow collector service configuration hierarchy. For a full configuration example, see "Example: Configuring a Flow Collector Interface on an M, MX or T Series Router" on page 203.

```
[edit]
chassis {
    fpc fpc-slot {
        pic pic-slot {
            monitoring-services {
                application flow-collector;
            }
        }
    }
}
interfaces {
```

```
    cp-fpc/pic/port {
        description ”flow_collector_interface”;
        unit 0 {
            family inet {
                address ip-address {
                    destination ip-address;
                }
            }
        }
        unit 1 {
            family inet {
                address ip-address {
                    destination ip-address;
                }
            }
        }
        unit 2 {
            family inet {
                address ip-address {
                    destination ip-address;
                }
            }
        }
    }
    interface-fpc/pic/port {
        description “export_interface”;
        unit 0 {
            family inet {
                address ip-address;
            }
        }
    }
    mo-fpc/pic/port  {
        description “monitoring_services_interface”;
        unit 0 {
            family inet;
        }
    }
    Ethernet-based-interface-fpc/pic/port {
        description “ input_interface”;
        encapsulation encapsulation-type;
    }
}
```

```
forwarding-options {
    monitoring group1 {
        family inet {
            output {
                export-format cflowd-version-5;
                flow-active-timeout value;
                flow-inactive-timeout value;
                flow-export-destination collector-pic;
                interface mo-fpc/pic/port {
                    source-address ip-address;
                }
            }
        }
    }
}
services {
    flow-collector {
        analyzer-address ip-address;
        analyzer-id name;
        retry value;
        retry-delay seconds;
        destinations {
            "ftp://username@ftp-server-address-1//directory/" {
                password "encrypted-password";
            }
            "ftp://username@ftp-server-address-2//directory/" {
                password "encrypted-password";
            }
        }
        file-specification {
            file-specification-name {
            }
            data-format flow-compressed;
            transfer timeout value record-level size;
        }
    }
    interface-map {
        file-specification file-specification-name;
         collector cp-fpc/pic/port;
        interface-name {
            file-specification file-specification-name;
            collector cp-fpc/pic/port;
        }
```

```
    }
    transfer-log-archive {
        filename-prefix filename;
        maximum-age timeout-value;
        archive-sites {
            "ftp://username@ip-address//directory/" {
                password "encrypted-password";
            }
        }
    }
}
```

Example: Configuring a Flow Collector Interface on an MX Series Router

IN THIS SECTION

Figure 23: Flow Collector Interface Topology Diagram

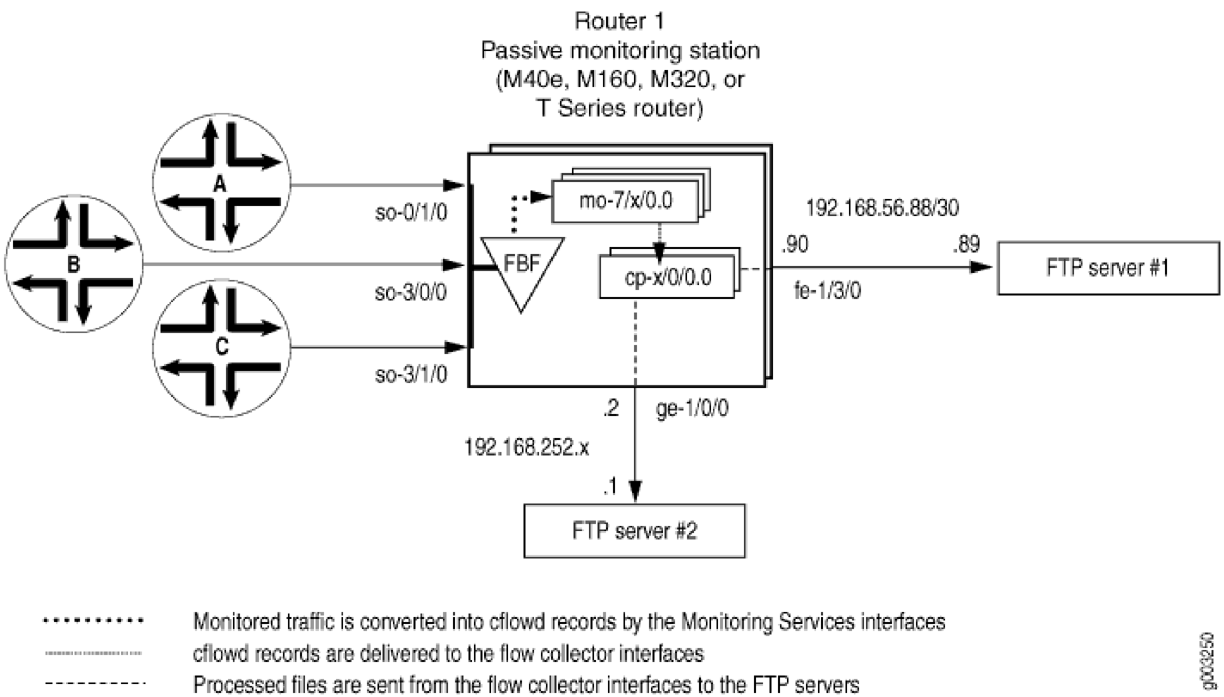


Figure 23 on page 204 shows the path traveled by monitored traffic as it passes through the router. Packets arrive at input interfaces **so-0/1/0**, **so-3/0/0**, and **so-3/1/0**. The raw packets are directed into a filter-based forwarding routing instance and processed into flow records by the monitoring services interfaces **mo-7/1/0**, **mo-7/2/0**, and **mo-7/3/0**. The flow records are compressed into files at the flow collector interfaces **cp-6/0/0** and **cp-7/0/0** and sent to the FTP server for analysis. Finally, a mandatory class-of-service (CoS) configuration is applied to export channels 0 and 1 on the flow collector interfaces to manage the outgoing processed files.

Router 1

```
[edit]
chassis {
    fpc 6 {
        pic 0 {
            monitoring-services {
                application flow-collector; # This converts a Monitoring Services II PIC
            }          # into a flow collector interface.
        }
    }
    fpc 7 {
        pic 0 {
            monitoring-services {
                application flow-collector; # This converts a Monitoring Services II PIC
            }             # into a flow collector interface.
        }
    }
}
interfaces {
    cp-6/0/0 {
                           unit 0 {# Logical interface .0 on a flow collector interface is export
            family inet { # channel 0 and sends records to the FTP server.
                filter {
                                           output cp-ftp; # Apply the CoS filter here.
                }
                address 10.0.0.1/32 {
                    destination 10.0.0.2;
                }
            }
        }
                           unit 1 { # Logical interface .1 on a flow collector interface is
export
            family inet { # channel 1 and sends records to the FTP server.
```

```
                filter {
                                    output cp-ftp; # Apply the CoS filter here.
                }
                address 10.1.1.1/32 {
                    destination 10.1.1.2;
                }
            }
        }
                    unit 2 { # Logical interface .2 on a flow collector interface is the
flow
            family inet { # receive channel that communicates with the Routing Engine.
                address 10.2.2.1/32 { # Do not apply a CoS filter on logical interface .2.
                    destination 10.2.2.2;
                }
            }
        }
    }
    cp-7/0/0 {
                    unit 0 { # Logical interface .0 on a flow collector interface is
export
            family inet { # channel 0 and sends records to the FTP server.
                filter {
                                    output cp-ftp; # Apply the CoS filter here.
                }
                address 10.3.3.1/32 {
                    destination 10.3.3.2;
                }
            }
        }
                    unit 1 { # Logical interface .1 on a flow collector interface is
export
            family inet { # channel 1 and sends records to the FTP server.
                filter {
                                    output cp-ftp; # Apply the CoS filter here.
                }
                address 10.4.4.1/32 {
                    destination 10.4.4.2;
                }
            }
        }
                    unit 2 { # Logical interface .2 on a flow collector interface is the
flow
            family inet { # receive channel that communicates with the Routing Engine.
```

```
                address 10.5.5.1/32 { # Do not apply a CoS filter on logical interface .2.
                    destination 10.5.5.2;
                }
            }
        }
    }
                     fe-1/3/0 { # This is the exit interface leading to the first FTP server.
        unit 0 {
            family inet {
                address 192.168.56.90/30;
            }
        }
    }
                     ge-1/0/0 { # This is the exit interface leading to the second FTP server.
        unit 0 {
            family inet {
                address 192.168.252.2/24;
            }
        }
    }
                     mo-7/1/0 {  # This is the first interface that creates flow records.
        unit 0 {
            family inet;
        }
    }
                     mo-7/2/0 { # This is the second interface that creates flow records.
        unit 0 {
            family inet;
        }
    }
                     mo-7/3/0 { # This is the third interface that creates flow records.
        unit 0 {
            family inet;
        }
    }
                     so-0/1/0 { # This is the first input interface that receives traffic to be
monitored.
        encapsulation ppp;
        unit 0 {
                               passive-monitor-mode; # This allows the interface to be passively
monitored.
            family inet {
                filter {
```

```
                                          input catch; # The filter-based forwarding filter is 
applied here.
                }
            }
        }
    }
                     so-3/0/0 { # This is the second interface that receives traffic to be 
monitored.
        encapsulation ppp;
        unit 0 {
                               passive-monitor-mode; # This allows the interface to be passively 
monitored.
            family inet {
                filter {
                                           input catch; # The filter-based forwarding filter is 
applied here.
                }
            }
        }
    }
                     so-3/1/0 { # This is the third interface that receives traffic to be 
monitored.
        encapsulation ppp;
        unit 0 {
                               passive-monitor-mode; # This allows the interface to be passively 
monitored.
            family inet {
                filter {
                                           input catch; # The filter-based forwarding filter is 
applied here.
                }
            }
        }
    }
}
forwarding-options {
                     monitoring group1 { # Always define your monitoring group here.
        family inet {
            output {
                export-format cflowd-version-5;
                flow-active-timeout 60;
                flow-inactive-timeout 15;
                                     flow-export-destination collector-pic; # Sends records to
```

```
the flow collector.
                interface mo-7/1/0.0 {
                    source-address 192.168.252.2;
                }
                interface mo-7/2/0.0 {
                    source-address 192.168.252.2;
                }
                interface mo-7/3/0.0 {
                    source-address 192.168.252.2;
                }
            }
        }
    }
}
routing-options {
    interface-routes {
        rib-group inet common;
    }
    rib-groups {
        common {
            import-rib [ inet.0 fbf_instance.inet.0 ];
        }
    }
    forwarding-table {
        export pplb;
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
class-of-service { # A class-of-service configuration for the flow collector interface
    interfaces { # is mandatory when implementing flow collector services.
                           cp-6/0/0 {
            scheduler-map cp-map;
        }
                           cp-7/0/0 {
            scheduler-map cp-map;
        }
    }
```

```
    scheduler-maps {
        cp-map {
            forwarding-class best-effort scheduler Q0;
            forwarding-class expedited-forwarding scheduler Q1;
            forwarding-class network-control scheduler Q3;
        }
    }
    schedulers {
        Q0 {
            transmit-rate remainder;
            buffer-size percent 90;
        }
        Q1 {
            transmit-rate percent 5;
            buffer-size percent 5;
            priority strict-high;
        }
        Q3 {
            transmit-rate percent 5;
            buffer-size percent 5;
        }
    }
}
firewall {
    family inet {
        filter cp-ftp { # This filter provides CoS for flow collector interface traffic.
            term t1 {
                then forwarding-class expedited-forwarding;
            }
        }
    }
    filter catch { # This firewall filter sends incoming traffic into the
        interface-specific; # filter-based forwarding routing instance.
        term def {
            then {
                count counter;
                routing-instance fbf_instance;
            }
        }
    }
}
routing-instances {
                     fbf_instance { # This instance sends traffic to the monitoring services
```

```
interface.
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop mo-7/1/0.0;
            }
        }
    }
}
services {
                     flow-collector { # Define properties for flow collector interfaces here.
                         analyzer-address 10.10.10.1; # This is the IP address of the analyzer.
                         analyzer-id server1; # This helps to identify the analyzer.
                         retry 3; # Maximum number of attempts by the PIC to send a file
transfer log.
                         retry-delay 30; # The time interval between attempts to send a file
transfer log.
                           destinations { # This defines the FTP servers that receive flow
collector output.
                                 "ftp://user@192.168.56.89//tmp/collect1/" {  # The primary FTP
server.
                password "$ABC123"; # SECRET-DATA
            }
                                 "ftp://user@192.168.252.1//tmp/collect2/" { # The second FTP
server.
                password "$ABC123"; # SECRET-DATA
            }
        }
                           file-specification {  # Define sets of flow collector characteristics
here.
                                 def-spec {
            }
                               data-format flow-compressed; # The default compressed output
format.
        }
                           f1 {
            name-format "cFlowd-py69Ni69-0-%D_%T-%I_%N.bcp.bi.gz";
                               data-format flow-compressed; # The default compressed output
format.
                               transfer timeout 1800 record-level 1000000; # Here are configured
values.
        }
    }
```

```
                    interface-map { # Allows you to map interfaces to flow collector interfaces.
                        file-specification def-spec; # Flows generated for default traffic are
sent to the
                        collector cp-7/0/0; # default flow collector interface cp-7/0/0.
                          so-0/1/0.0 {# Flows generated for the so-0/1/0 interface are sent
                              collector cp-6/0/0; # to cp-6/0/0, and the file-specification
used is “default”.
        }
                          so-3/0/0.0 { # Flows generated for the so-3/0/0 interface are sent
                              file-specification f1; # to cp-6/0/0, and the file-specification
used is "f1."
                              collector cp-6/0/0;
        }
                        so-3/1/0.0; # Because no settings are defined, flows generated for this
    }
                    transfer-log-archive { # Sends flow collector interface log files to an FTP
server.
        filename-prefix so_3_0_0_log;
        maximum-age 15;
        archive-sites {
            "ftp://user@192.168.56.89//tmp/transfers/" {
                password "$ABC123";
            }
        }
    }
}
```

Verifying Your Work

To verify that your flow collector configuration is working, use the following commands on the monitoring station that is configured for flow collection:

- `clear services flow-collector statistics`
- `request services flow-collector change-destination (primary | secondary)`
- `request services flow-collector test-file-transfer`
- `show services flow-collector file interface (detail | extensive | terse)`
- `show services flow-collector (detail | extensive)`
- `show services flow-collector input interface (detail | extensive | terse)`

The following section shows the output of the `show` commands used with the configuration example:

```
user@router1> show services flow-collector input interface cp-6/0/0 detail
Interface                      Packets        Bytes
mo-7/1/0.0                        6170      8941592

user@router1>  show services flow-collector interface all detail
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
  Packets     Bytes     Flows Uncompressed  Compressed     FTP bytes FTP files
                                     Bytes       Bytes
     6736   9757936    195993     21855798     3194148             0         0
Flow collector interface: cp-7/0/0
Interface state: Collecting flows
  Packets     Bytes     Flows Uncompressed  Compressed     FTP bytes FTP files
                                     Bytes       Bytes
        0         0         0            0           0             0         0

user@router1>  show services flow-collector input interface cp-6/0/0 extensive
Interface                      Packets        Bytes
mo-7/1/0.0                        6260      9074096

user@router1>  show services flow-collector interface cp-6/0/0 extensive
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Memory:
    Used: 19593212, Free: 479528656
Input:
    Packets: 6658, per second: 0, peak per second: 0
    Bytes: 9647752, per second: 12655, peak per second: 14311
    Flow records processed: 193782, per second: 252, peak per second: 287
Allocation:
    Blocks allocated: 174, per second: 0, peak per second: 0
    Blocks freed: 0, per second: 0, peak per second: 0
    Blocks unavailable: 0, per second: 0, peak per second: 0
Files:
    Files created: 1, per second: 0, peak per second: 0
    Files exported: 0, per second: 0, peak per second: 0
    Files destroyed: 0, per second: 0, peak per second: 0
Throughput:
    Uncompressed bytes: 21075152, per second: 52032, peak per second: 156172
    Compressed bytes: 3079713, per second: 7618, peak per second: 22999
```

```
Packet drops:
    No memory: 0, Not IP: 0
    Not IPv4: 0, Too small: 0
    Fragments: 0, ICMP: 0
    TCP: 0, Unknown: 0
    Not JUNOS flow: 0
File Transfer:
    FTP bytes: 0, per second: 0, peak per second: 0
    FTP files: 0, per second: 0, peak per second: 0
    FTP failure: 0
Export channel: 0
    Current server: Secondary
    Primary server state: OK, Secondary server state: OK
Export channel: 1
    Current server: Secondary
    Primary server state: OK, Secondary server state: OK

user@router1>  show services flow-collector file interface cp-6/0/0 terse
File name                                                         Flows State
cFlowd-py69Ni69-0-20031112_014301-so_3_0_0_0.bcp.bi.gz          185643 Active

user@router1>  show services flow-collector file interface cp-6/0/0 detail
Filename: cFlowd-py69Ni69-0-20031112_014301-so_3_0_0_0.bcp.bi.gz
  Throughput:
    Flow records: 187067, Uncompressed bytes: 21121960, Compressed bytes: 2965643
  Status:
    State: Active, Transfer attempts: 0

user@router1>  show services flow-collector file interface cp-6/0/0 extensive
Filename: cFlowd-py69Ni69-0-20031112_014301-so_3_0_0_0.bcp.bi.gz
  Throughput:
    Flow records: 188365, per second: 238, peak per second: 287
    Uncompressed bytes: 21267756, per second: 27007, peak per second: 32526
    Compressed bytes: 2965643, per second: 0, peak per second: 22999
  Status:
    Compressed blocks: 156, Block count: 156
    State: Active, Transfer attempts: 0
```

To clear statistics for a flow collector interface, issue the `clear services flow-collector statistics interface (all | interface-name)` command.

Another useful flow collector option allows you to change the FTP server from primary to secondary and test for FTP transfers. To force the flow collector interface to use a primary or secondary FTP server, include the **primary** or **secondary** option when you issue the `request services flow-collector change-destination interface cp-`*`fpc/pic/port`* command.

If you configure only one primary server and issue this command with the **primary** option, you receive the error message "Destination change not needed." If the secondary server is not configured and you issue this command with the **secondary** option, you receive the error message "Destination not configured." Otherwise, when both servers are configured properly, successful output appears as follows.

```
user@router1> request services flow-collector change-destination interface           cp-6/0/0
primary
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Destination change successful

user@router1>  request services flow-collector change-destination interface  cp-6/0/0 secondary
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Destination change successful
```

Other options for the `request services flow-collector change-destination interface cp-`*`fpc/pic/port`* command are **immediately** (which forces an instant switchover), **gracefully** (the default behavior that allows a gradual switchover), **clear-files** (which purges existing data files), and **clear-logs** (which purges existing log files).

To verify that transfer log files are being scheduled for delivery to the FTP servers, issue the `request services flow-collector test-file-transfer` *`filename`* `interface cp-`*`fpc/pic/port`* command. Include the desired export channel (zero or one) and target FTP server (primary or secondary) with this command.

```
user@router1> request services flow-collector test-file-transfer test_file           interface
cp-6/0/0 channel-one primary
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Response: Test file transfer successfully scheduled
```

Another way you can check for the success of your file transfers is by analyzing the transfer log. A transfer log sends detailed information about files that are collected and processed by the flow collector interface. Table 32 on page 216 explains the various fields available in the transfer log.

Table 32: Flow Collector Interface Transfer Log Fields

Field	Explanation
fn	Filename
sz	File size
nr	Number of records
ts	Timestamp with the format of year (4 digits), month (2 digits), day (2 digits), hours (2 digits), minutes (2 digits), and seconds (2 digits).
sf	Success flag—The values are **1** for success and **0** for failure.
ul	Server URL
rc	FTP result code
er	FTP error text
tt	Transfer time

This is an example of a successful transfer log:

```
fn="cFlowd-py69Ni69-0-20040227_230438-at_4_0_0_4_3.bcp.bi.gz":sz=552569
:nr=20000:ts="20040227230855":sf=1:ul="ftp://10.63.152.1/tmp/server1/:"rc=250:
er="":tt=3280
```

This is an example of a transfer log when an FTP session fails:

```
fn="cFlowd-py69Ni69-0-20040227_230515-at_4_0_0_2_8.bcp.bi.gz":sz=560436
:nr=20000:ts="20040227230855":sf=1:ul="ftp://10.63.152.1/tmp/server1/:"rc=250
:er="":tt=3290
```

As the flow collector interface receives and processes flow records, the PIC services logging process (fsad) handles the following tasks:

- When the flow collector interface transfers a file to the FTP server, a temporary log file is created in the **/var/log/flowc** directory. The temporary log file has this filenaming convention:

  <*hostname*>_<*filename_prefix*>_ *YYYYMMDD_hhmmss*.tmp

  hostname is the hostname of the transfer server, ***filename_prefix*** is the same value defined with the `filename-prefix` statement at the [**edit services flow-collector transfer-log-archive**] hierarchy level, ***YYYYMMDD*** is the year, month, and date, and ***hhmmss*** is the timestamp indicating hours, minutes, and seconds.

- After the log file has been stored in the router for the length of time specified by the `maximum-age` statement at the [**edit services flow-collector transfer-log-archive**] hierarchy level (the default is 120 minutes), the temporary log file is converted to an actual log file and the temporary file is deleted. The new log file retains the same naming conventions, except the extension is **.log*.

- When the final log file is created and compressed, the PIC services logging process (fsad) tries to send the log file from the **/var/log/flowc** directory to an FTP server. You can specify up to five FTP servers to receive the log files by including the `archive-sites` statement at the [**edit services flow-collector transfer-log-archive**] hierarchy level. The logging process attempts to send the log file to one server at a time, in order of their appearance in the configuration. Upon the first successful transfer, the log file is deleted and the logging process stops sending log files to the remaining FTP servers in the list.

- If the log file transfer is not successful, the log file is moved to the **/var/log/flowc/failed** directory. Every 30 minutes, the logging process tries to resend the log files. After the log files are transferred successfully, they are deleted from the **/var/log/flowc/failed** directory.

  NOTE: If the memory for a flow collector interface is full, the interface might drop incoming packets.

After the flow collector interface successfully delivers the processed information file to the FTP server, you can analyze the file. The file contains detailed information about the flows collected and processed by the flow collector interface. Table 33 on page 218 explains the various fields available in the flow collector interface file.

Table 33: Flow Collector Interface File Fields in Order of Appearance

Field	Explanation
linkDir	Link directory—A randomly generated number used to identify the record
analyzer-address	Analyzer address
analyzer-ID	Analyzer identifier
ifAlias	Interface identifier
source-address	Source address
destination-address	Destination address
packets	Number of packets
bytes	Number of bytes
start-time	Start time
end-time	End time
source-port	Source port
destination-port	Destination port
tcp_flag	TCP flag
protocol	IP protocol number
src_AS_number	Source AS number

Table 33: Flow Collector Interface File Fields in Order of Appearance *(Continued)*

Field	Explanation
dst_AS_number	Destination AS number

This is an example of output from a flow collector interface file:

```
11799241612374557782|10.10.10.1|server1|at_4_0_0_4|192.168.10.100|10.0.0.1|8|
3136|1077926402|1077926402|8224|12336|27|6|0|0
```

CHAPTER 4

Processing and Exporting Multiple Records Using Flow Collection

IN THIS CHAPTER

Flow Collection Overview

You can process and export multiple cflowd records with a flow collector interface. You create a flow collector interface on a Monitoring Services II or Multiservices 400 PIC. The flow collector interface combines multiple cflowd records into a compressed ASCII data file and exports the file to an FTP server. To convert a services PIC into a flow collector interface, include the `flow-collector` statement at the `[edit chassis fpc` *`fpc-slot`* `pic` *`pic-slot`* `monitoring-services application]` hierarchy level.

You can use the services PIC for either flow collection or monitoring, but not for both types of service simultaneously. When converting the PIC between service types, you must configure the `flow-collector` statement, take the PIC offline, and then bring the PIC back online. Restarting the router does not enable the new service type.

A flow collector interface, designated by the **cp-*fpc/pic/port*** interface name, requires three logical interfaces for correct operation. Units 0 and 1 are used to send the compressed ASCII data files to an FTP server, while Unit 2 is used to receive cflowd records from a monitoring services interface.

NOTE: Unlike conventional interfaces, the `address` statement at the `[edit interfaces cp-`*`fpc/pic/port`* `unit` *`unit-number`* `family inet]` hierarchy level corresponds to the IP address of the Routing Engine. Likewise, the `destination` statement at the `[edit interfaces cp-`*`fpc/pic/`*

port unit *unit-number* family inet address *ip-address*] hierarchy level corresponds to the IP address of the flow collector interface. As a result, you must configure the destination statement for Unit 0 and 1 with *local* addresses that can reach the FTP server. Similarly, configure the destination statement for Unit 2 with a *local* IP address so it can reach the monitoring services interface that sends cflowd records.

To activate flow collector services after the services PIC is converted into a flow collector, include the flow-collector statement at the [edit services] hierarchy level.

After you activate the flow collector, you need to configure the following components:

- Destination of the FTP server
- File specifications
- Input interface-to-flow collector interface mappings
- Transfer log settings

RELATED DOCUMENTATION

Configuring Flow Collection

IN THIS SECTION

Configuring Destination FTP Servers for Flow Records

Flow collection destinations are where the compressed ASCII data files are sent after the cflowd records are collected and processed. To specify the destination FTP server, include the `destinations` statement at the `[edit services flow-collector]` hierarchy level. You can specify up to two FTP server destinations and include the password for each configured server. If two FTP servers are configured, the first server in the configuration is the primary server and the second is a backup server.

To configure a destination for flow collection files, include the `destinations` statement at the `[edit services flow-collector]` hierarchy level:

```
[edit services flow-collector]
destinations {
    ftp:url {
        password "password";
    }
}
```

To specify the destination FTP server, include the `ftp:url` statement. The value *url* is the FTP server address for the primary flow collection destination and can include macros.

When you include macros in the `ftp:url` statement, a directory can be created only for a single level. For example, the path `ftp://10.2.2.2/%m/%Y` expands to `ftp://10.2.2.2/01/2005`, and the software attempts to create the directory `01/2005` on the destination FTP server. If the `01/` directory already exists on the destination FTP server, the software creates the **/2005/ directory** one level down. If the `01/` directory does not exist on the destination FTP server, the software cannot create the **/2005/ directory**, and the FTP server destination fails. For more information about macros, see *ftp*.

To specify the FTP server password, include the `password` "*password*" statement. The password must be enclosed in quotation marks. You can specify up to two destination FTP servers. The first destination specified is considered the primary destination.

Configuring a Packet Analyzer

You can specify values for the IP address and identifier of a packet analyzer to which the flow collector interface sends traffic for analysis. The values you specify here override any default values configured elsewhere.

To configure an IP address and identifier for the packet analyzer, include the `analyzer-address` and `analyzer-id` statements at the `[edit services flow-collector]` hierarchy level:

```
[edit services flow-collector]
analyzer-address address;
analyzer-id name;
```

Configuring File Formats

You configure data file formats, name formats, and transfer characteristics for the flow collection files. File records are sent to the destination FTP server when the timer expires or when a preset number of records are received, whichever comes first.

To configure the flow collection file format, include the `file-specification` statement at the `[edit services flow-collector]` hierarchy level:

```
[edit services flow-collector]
file-specification {
    variant variant-number {
        data-format format;
        name-format format;
        transfer {
            record-level number;
            timeout seconds;
        }
    }
}
```

To set the data file format, include the `data-format` statement. To set the file name format, include the `name-format` statement. To set the export timer and file size thresholds, include the `transfer` statement and specify values for the `timeout` and `record-level` options.

For example, you can specify the name format as follows:

```
[edit services flow-collector file-specification variant variant-number]
name-format "cFlowd-py69Ni69-0-%D_%T-%I_%N.bcp.bi.gz";
```

In this example, `cFlowd-py69Ni69-0` is the static portion used verbatim, `%D` is the date in YYYYMMDD format, `%T` is the time in HHMMSS format, `%I` is the value of `ifAlias`, `%N` is the generation number, and bcp.bi.gz is a user-configured string. A number of macros are supported for expressing the date and time information in different ways; for a complete list, see the summary section for *name-format*.

Configuring Interface Mappings

You can match an input interface with a flow collector interface and apply the preset file specifications to the input interface.

To configure an interface mapping, include the `interface-map` statement at the `[edit services flow-collector]` hierarchy level:

```
[edit services flow-collector]
interface-map {
    collector interface-name;
    file-specification variant-number;
    interface-name {
        collector interface-name;
        file-specification variant-number;
    }
}
```

To configure the default flow collector and file specifications for all input interfaces, include the `file-specification` and `collector` statements at the `[edit services flow-collector interface-map]` hierarchy level. To override the default settings and apply flow collector and file specifications to a specific input interface, include the `file-specification` and `collector` statements at the `[edit services flow-collector interface-map` *`interface-name`*`]` hierarchy level.

Configuring Transfer Logs

You can configure the filename, export interval, maximum size, and destination FTP server for log files containing the transfer activity history for a flow collector interface.

To configure a transfer log, include the `transfer-log-archive` statement at the `[edit services flow-collector]` hierarchy level:

```
[edit services flow-collector]
transfer-log-archive {
    archive-sites {
        ftp:url {
            password "password";
            username username;
```

```
        }
    }
    filename-prefix prefix;
    maximum-age minutes;
}
```

To configure the destination for archiving files, include the `archive-sites` statement. Specify the filename as follows:

```
[edit services flow-collector transfer-log]
filename "cFlowd-py69Ni69-0-%D_%T";
```

where `cFlowd-py69Ni69-0` is the static portion used verbatim, `%D` is the date in YYYYMMDD format, and `%T` is the time in HHMMSS format.

You can optionally include the following statements:

- `filename-prefix`—Sets a standard prefix for all the logged files.
- `maximum-age`—Specifies the duration a file remains on the server. The range is 1 through 360 minutes.

Configuring Retry Attempts

You can specify values for situations in which the flow collector interface needs more than one attempt to transfer log files to the FTP server:

- Maximum number of retry attempts
- Amount of time the flow collector interface waits between successive retries

To configure retry settings, include the `retry` and `retry-delay` statements at the `[edit services flow-collector]` hierarchy level:

```
retry number;
retry-delay seconds;
```

The `retry` value can be from 0 through 10. The `retry-delay` value can be from 0 through 60 seconds.

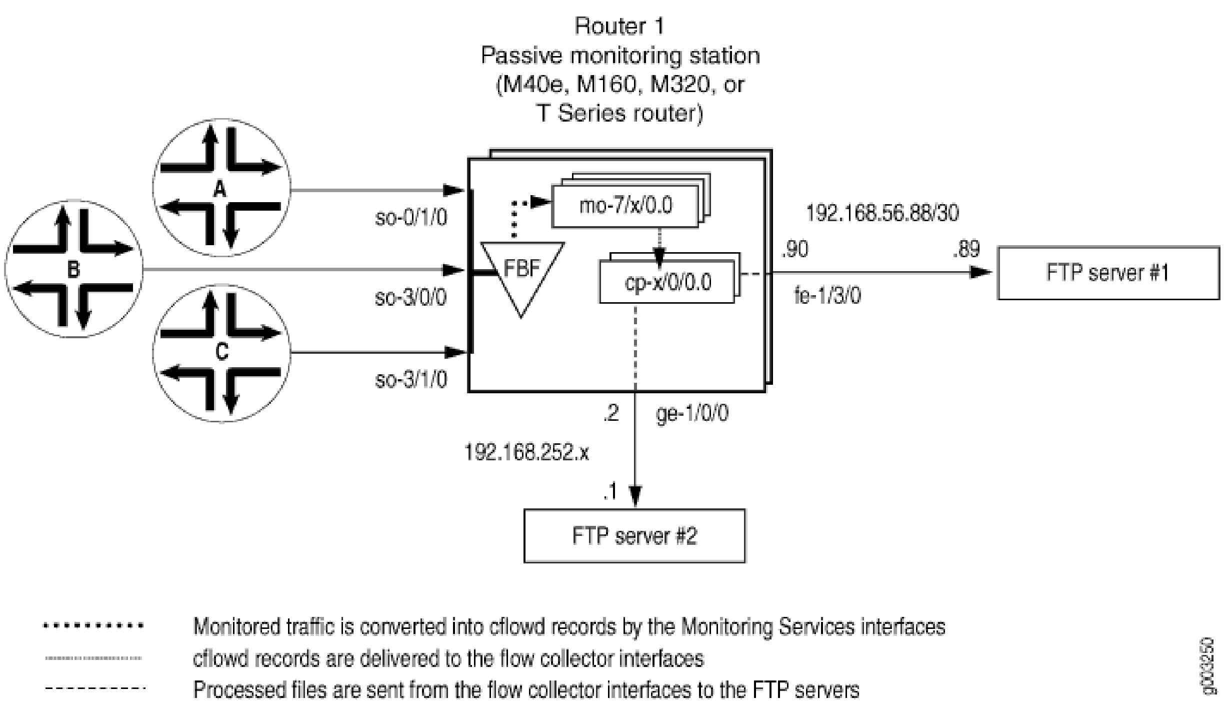
RELATED DOCUMENTATION

Flow Collection Overview | 220
Sending cflowd Records to Flow Collector Interfaces | 234

Example: Configuring Flow Collection

Figure 24 on page 226 shows the path traveled by monitored traffic as it passes through the router. Packets arrive at input interfaces so-0/1/0, so-3/0/0, and so-3/1/0. The raw packets are directed into a filter-based forwarding routing instance and processed into cflowd records by the monitoring services interfaces mo-7/1/0, mo-7/2/0, and mo-7/3/0. The cflowd records are compressed into files at the flow collector interfaces cp-6/0/0 and cp-7/0/0 and sent to the FTP server for analysis. Finally, a mandatory class-of-service (CoS) configuration is applied to export channels 0 and 1 on the flow collector interfaces to manage the outgoing processed files.

Figure 24: Flow Collector Interface Topology Diagram




```
 [edit]
chassis {
    fpc 6 {
        pic 0 {
```

```
            monitoring-services {
                application flow-collector; # This converts a Monitoring Services II or
                                   # Multiservices 400 PIC into a flow collector interface.
            }
        }
    }
    fpc 7 {
        pic 0 {
            monitoring-services {
                application flow-collector; # This converts a Monitoring Services II or
                                   # Multiservices 400 PIC into a flow collector interface.
            }
        }
    }
}
interfaces {
    cp-6/0/0 {
        unit 0 { # Logical interface .0 on a flow collector interface is export
            family inet { # channel 0 and sends records to the FTP server.
                filter {
                    output cp-ftp; # Apply the CoS filter here.
                }
                address 10.0.0.1/32 {
                    destination 10.0.0.2;
                }
            }
        }
        unit 1 { # Logical interface .1 on a flow collector interface is export
            family inet {# channel 1 and sends records to the FTP server.
                filter {
                    output cp-ftp; # Apply the CoS filter here.
                }
                address 10.1.1.1/32 {
                    destination 10.1.1.2;
                }
            }
        }
        unit 2 { # Logical interface .2 on a flow collector interface is the flow
            family inet { # receive channel that communicates with the Routing Engine.
                address 10.2.2.1/32 { # Do not apply a CoS filter on logical interface .2.
                    destination 10.2.2.2;
                }
            }
```

```
        }
    }
    cp-7/0/0 {
        unit 0 {# Logical interface .0 on a flow collector interface is export
            family inet {# channel 0 and sends records to the FTP server.
                filter {
                    output cp-ftp;# Apply the CoS filter here.
                }
                address 10.3.3.1/32 {
                    destination 10.3.3.2;
                }
            }
        }
        unit 1 {# Logical interface .1 on a flow collector interface is export
            family inet {# channel 1 and sends records to the FTP server.
                filter {
                    output cp-ftp;# Apply the CoS filter here.
                }
                address 10.4.4.1/32 {
                    destination 10.4.4.2;
                }
            }
        }
        unit 2 {# Logical interface .2 on a flow collector interface is the flow
            family inet {# receive channel that communicates with the Routing Engine.
                address 10.5.5.1/32 {# Do not apply a CoS filter on logical interface .2.
                    destination 10.5.5.2;
                }
            }
        }
    }
    fe-1/3/0 { # This is the exit interface leading to the first FTP server.
        unit 0 {
            family inet {
                address 192.168.56.90/30;
            }
        }
    }
    ge-1/0/0 { # This is the exit interface leading to the second FTP server.
        unit 0 {
            family inet {
                address 192.168.252.2/24;
            }
```

```
        }
    }
    mo-7/1/0 { # This is the first interface that creates cflowd records.
        unit 0 {
            family inet;
        }
    }
    mo-7/2/0 { # This is the second interface that creates cflowd records.
        unit 0 {
            family inet;
        }
    }
    mo-7/3/0 { # This is the third interface that creates cflowd records.
        unit 0 {
            family inet;
        }
    }
    so-0/1/0 { # This is the first input interface that receives traffic to be monitored.
        encapsulation ppp;
        unit 0 {
            passive-monitor-mode; # This allows the interface to be passively monitored.
            family inet {
                    filter {
                        input catch; # The filter-based forwarding filter is applied here.
                    }
            }
        }
    }
    so-3/0/0 { # This is the second interface that receives traffic to be monitored.
        encapsulation ppp;
        unit 0 {
            passive-monitor-mode; # This allows the interface to be passively monitored.
            family inet {
                    filter {
                        input catch; # The filter-based forwarding filter is applied here.
                    }
            }
        }
    }
so-3/1/0 { # This is the third interface that receives traffic to be monitored.
    encapsulation ppp;
    unit 0 {
        passive-monitor-mode; # This allows the interface to be passively monitored.
```

```
        family inet {
            filter {
                input catch; # The filter-based forwarding filter is applied here.
            }
        }
    }
}
forwarding-options {
    monitoring group1 {# Always define your monitoring group here.
        family inet {
            output {
                export-format cflowd-version-5;
                flow-active-timeout 60;
                flow-inactive-timeout 15;
                flow-export-destination collector-pic;  # Sends records to the flow collector.
                interface mo-7/1/0.0 {
                    source-address 192.168.252.2;
                }
                interface mo-7/2/0.0 {
                    source-address 192.168.252.2;
                }
                interface mo-7/3/0.0 {
                    source-address 192.168.252.2;
                }
            }
        }
}
firewall {
    family inet {
        filter cp-ftp { # This filter provides CoS for flow collector interface traffic.
            term t1 {
                then forwarding-class expedited-forwarding;
            }
        }
    }
    filter catch { # This firewall filter sends incoming traffic into the
        interface-specific;# filter-based forwarding routing instance.
            term def {
                then {
                    count counter;
                    routing-instance fbf_instance;
                }
            }
```

```
        }
    }
routing-options {
    interface-routes {
        rib-group inet common;
    }
    rib-groups {
        common {
            import-rib [inet.0 fbf_instance.inet.0];
        }
    }
    forwarding-table {
        export pplb;
    }
}
policy-options {
    policy-statement pplb {
        then {
            load-balance per-packet;
        }
    }
}
routing-instances {
    fbf_instance { # This instance sends traffic to the monitoring services interface.
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop mo-7/1/0.0;
            }
        }
    }
}
class-of-service { # A class-of-service configuration for the flow collector interface
    interfaces { # is required for flow collector services.
        cp-6/0/0 {
            scheduler-map cp-map;
        }
        cp-7/0/0 {
            scheduler-map cp-map;
        }
    }
}
scheduler-maps {
```

```
    cp-map {
        forwarding-class best-effort scheduler Q0;
        forwarding-class expedited-forwarding scheduler Q1;
        forwarding-class network-control scheduler Q3;
    }
}
schedulers {
    Q0 {
        transmit-rate remainder;
        buffer-size percent 90;
    }
    Q1 {
        transmit-rate percent 5;
        buffer-size percent 5;
        priority strict-high;
    }
    Q3 {
        transmit-rate percent 5;
        buffer-size percent 5;
    }
}
services {
    flow-collector { # Define properties for flow collector interfaces here.
        analyzer-address 10.10.10.1; # This is the IP address of the analyzer.
        analyzer-id server1; # This helps to identify the analyzer.
        retry 3;   # Maximum number of attempts by the PIC to send a file transfer log.
        retry-delay 30; # The time interval between attempts to send a file transfer log.
        destinations { # This defines the FTP servers that receive flow collector output.
            "ftp://user@192.168.56.89//tmp/collect1/" { # The primary FTP server.
                password "$ABC123"; # SECRET-DATA
            }
            "ftp://user@192.168.252.1//tmp/collect2/" { # The secondary FTP server.
                password "$ABC123";  # SECRET-DATA
            }
        }
        file-specification { # Define sets of flow collector characteristics here.
            def-spec {
                name-format "default-allInt-0-%D_%T-%I_%N.bcp.bi.gz";
                data-format flow-compressed; # The default compressed output format.
            } # When no overrides are specified, a collector uses default transfer values.
            f1 {
                name-format "cFlowd-py69Ni69-0-%D_%T-%I_%N.bcp.bi.gz";
                data-format flow-compressed; # The default compressed output format.
```

```
                transfer timeout 1800 record-level 1000000; # Here are configured values.
            }
        }
        interface-map { # Allows you to map interfaces to flow collector interfaces.
            file-specification def-spec; # Flows generated for default traffic are sent to the
            collector cp-7/0/0; # default flow collector interface "cp-7/0/0".
            so-0/1/0.0 { # Flows generated for the so-0/1/0 interface are sent
                collector cp-6/0/0; # to cp-6/0/0, and the file-specification used is
            } # "default."
            so-3/0/0.0 { # Flows generated for the so-3/0/0 interface are sent
                file-specification f1; # to cp-6/0/0, and the file-specification used is "f1."
                collector cp-6/0/0;
            }
            so-3/1/0.0; # Because no settings are defined, flows generated for this
        } # interface use interface cp-7/0/0 and the default file specification.
        transfer-log-archive { # Sends flow collector interface log files to an FTP server.
            filename-prefix so_3_0_0_log;
            maximum-age 15;
            archive-sites {
                "ftp://user@192.168.56.89//tmp/transfers/" {
                    password "$ABC123";
                }
            }
        ]
    }
}
```

RELATED DOCUMENTATION

Flow Collection Overview | 220

Configuring Flow Collection | 221

Sending cflowd Records to Flow Collector Interfaces | 234

Configuring Flow Collection Mode and Interfaces on Router Services PICs on M and T Series Routers | 234

Sending cflowd Records to Flow Collector Interfaces

To specify a flow collector interface as the destination for cflowd records coming from a services PIC, include the `collector-pic` statement at the `[edit forwarding-options monitoring group-name family inet output flow-export-destination]` hierarchy level:

```
[edit forwarding-options monitoring group-name family inet output flow-export-destination]
collector-pic;
```

You can select either the flow collector interface or a cflowd server as the destination for cflowd records, but not both at the same time.

RELATED DOCUMENTATION

Flow Collection Overview | 220
Configuring Flow Collection | 221
Configuring Flow Collection Mode and Interfaces on Router Services PICs on M and T Series Routers | 234
Example: Configuring Flow Collection | 226

Configuring Flow Collection Mode and Interfaces on Router Services PICs on M and T Series Routers

You can select the services PIC to run in either flow collection mode or monitoring mode, but not both.

To set the services PIC to run in flow collection mode, include the `flow-collector` statement at the `[edit chassis fpc slot-number pic pic-number monitoring-services application]` hierarchy level:

```
[edit chassis fpc slot-number pic pic-number monitoring-services application]
flow-collector;
```

To specify flow collection interfaces, you configure the `cp` interface at the `[edit interfaces]` hierarchy level:

```
[edit interfaces]
cp-fpc/pic/port {
```

```
    ...
}
```

RELATED DOCUMENTATION

Flow Collection Overview | 220

Configuring Flow Collection | 221

Sending cflowd Records to Flow Collector Interfaces | 234

Example: Configuring Flow Collection | 226

CHAPTER 5

Logging Flow Monitoring Records with Version 9 and IPFIX Templates for NAT Events

IN THIS CHAPTER

Understanding NAT Event Logging in Flow Monitoring Format on an MX Series Router or NFX250

Starting with Junos OS Release 14.2R2 and 15.1R1, you can configure MX Series routers with MS-MPCs and MS-MICs to log network address translation (NAT) events using the Junos Traffic Vision (previously known as Jflow) version 9 or IPFIX (version 10) template format. You can also configure MX Series routers with MX-SPC3 services cards with this capability starting from Junos OS Release 19.3R2.

NAT event logger generates messages in flow monitoring format for various NAT events, such as the creation of a NAT entry, deletion of a NAT entry, and for invalid NAT processing (such as NAT address pools or address values being exhausted for allocation). These events also support NAT64 translations (translation of IPv6 addresses to IPv4 addresses), binding information base (BIB) events, and more detailed error generation. The generated records or logs for NAT events in flow template format are sent from the MS-MIC or MS-MPC or MX-SPC3 to the specified host or external device that functions as the NetFlow collector. This method of generating flow monitoring records for NAT events enables cohesive and streamlined analysis of NAT traffic and troubleshooting of NAT-related problems. You can enable the capability to send flow monitoring records for NAT operations to an external collector and the capability to use the system logging protocol (syslog) to generate session logging for different services at the same time.

The flow records and the templates are encapsulated in an UDP or IP packet and sent to the collector. However, TCP-based logging of monitoring records for NAT events is not supported. Carrier-grade NAT (CGN) devices are required to log events creation and deletion of translations and information about the resources it manages. Flow monitoring logs can be optionally configured in your network topology in addition to the system logging (syslog) capability, which causes logs to be saved from the PIC to either the in the **/var/log** directory of the Routing Engine (local) or to an external server (remote). Generally, flow collectors are the part of a vast network infrastructure containing several third-party devices, which perform various correlations and mappings with logs of other databases. Therefore, collection of NAT-related flow monitoring records as logs or template records is useful on the hosts or devices that function as collectors in an overall and comprehensive perspective. You can enable logging of flow monitoring records for NAT events at the service-set level to enable version 9 or IPFIX flow records to be generated as logs when NAT is configured on the router.

The NetFlow collector receives flow records in version 9 or IPFIX format from one or more exporters. It processes the received export packets by parsing and saving the flow record details. Flow records can be optionally aggregated before being stored on the hard disk. The NetFlow collector is also referred to as the collector. The exporter monitors packets entering an observation point and creates flows from these packets. The information from these flows is exported in the form of flow records to the NetFlow Collector. An observation point is a location in the network where IP packets can be overseen and monitored; for example, one or a set of interfaces on a network device such as a router. Every observation point is associated with an observation domain, which is a cluster of observation points, and constitutes the largest aggregatable set of flow information at the network device with NetFlow services enabled.

A FlowSet is a generic term for a collection of Flow Records that have a similar pattern or format. In an export packet, one or more FlowSets follow the packet header. A Template FlowSet comprises one or more template records that have been grouped together in an export packet. An Options Template FlowSet contains one or more Options Template records that are combined together in an export packet. A Data FlowSet is one or more records, of the same type, that are grouped together in an export packet. Each record is either a flow data record or an options data record that has been previously specified by a Template Record or an Options Template Record. One of the essential elements in the NetFlow format is the Template FlowSet. Templates vastly enhance the flexibility of the Flow Record format because they allow the collector to process Flow Records without necessarily knowing the interpretation of all the data in the Flow Record.

You can configure the capability to transmit records or log messages in version 9 and IPFIX traffic flow formats generated for NAT events to an external, off-box high-speed NetFlow collector for easy and effective monitoring and diagnosis of the logs. By default, this functionality is disabled. With a high number of NAT events, this mechanism of exporting logs to an external log collector might cause scaling considerations such as loss of a few flow records. To enable the mechanism to record logging messages in flow monitoring format for NAT events, you can now include the `jflow-log` statement at the `[edit services]` hierarchy level. You can configure a collector, which is an external host to which the flow monitoring formatted logs are sent, or a group of collectors. A group of collectors is useful in scenarios in which you want to combine a set of collector devices and define common settings for logging NAT events for all the collectors in the cluster or group.

To configure a collector and its parameters, such as the source IP address from which the records are sent and the destination address of the collector, include the `collector` *`collector-name`* statement and its substatements at the `[edit services jflow-log]` hierarchy level. To specify a collector group or a cluster, include the `collector-group` *`collector-group-name`* statement and its substatements at the `[edit services jflow-log]` hierarchy level.

You need to configure a template profile and associate it with the collector. The profile defines the characteristics of the flow monitoring record template, such as the version of flow monitoring (version 9 or IPFIX), the refresh rate, in either packets or seconds, and the type of service or application (NAT in this case) for which flow records must be sent to the collector. To specify a template profile, include the `template-profile` *`template-profile-name`* statement at the `[edit services jflow-log]` hierarchy level. To specify the maximum number of messages to be collected per second for NAT error events, include the `message-rate-limit` *`messages-per-second`* statement at the `[edit interfaces` *`ms-interface-name`* `service-options jflow-log]` hierarchy level.

Use of version 9 and IPFIX allows you to define a flow record template suitable for IPv4 traffic, IPv6 traffic, MPLS traffic, a combination of IPv4 and MPLS traffic, or peer AS billing traffic. Templates and the fields included in the template are transmitted to the collector periodically, and the collector need not be aware of the router configuration. You must define a template profile properties for a NAT service and associate the defined template profile with a service set to enable the flow monitoring log functionality for NAT events. To define the template profile characteristics for recording flow monitoring logs for NAT events, include the `template-profile` *`template-profile-name`* statement at the `[edit services jflow-log]` hierarchy level. To associate the template profile for recording flow monitoring logs for NAT events with a service-set level, which applies for all the services in the system, include the `template-profile` *`template-profile-name`* statement at the `[edit services service-set` *`service-set-name`*`]` hierarchy level.

To view statistical information on the logs generated in flow monitoring format for the interfaces and service sets configured on the system, use the `show services service-sets statistics jflow-log` command.

The following system log messages for various NAT events are logged using the system logging (syslog) capability:

- JSERVICES_SESSION_OPEN
- JSERVICES_SESSION_CLOSE
- JSERVICES_NAT_OUTOF_ADDRESSES
- JSERVICES_NAT_OUTOF_PORTS
- JSERVICES_NAT_RULE_MATCH
- JSERVICES_NAT_POOL_RELEASE
- JSERVICES_NAT_PORT_BLOCK_ALLOC
- JSERVICES_NAT_PORT_BLOCK_RELEASE
- JSERVICES_NAT_PORT_BLOCK_ACTIVE

The following NAT events are logged using the flow monitoring log capability using version 9 and IPFIX flow templates:

- NAT44 session create
- NAT44 session delete
- NAT addresses exhausted
- NAT64 session create
- NAT64 session delete
- NAT44 BIB create
- NAT44 BIB delete
- NAT64 BIB create
- NAT64 BIB delete
- NAT ports exhausted
- NAT quota exceeded
- NAT Address binding create
- NAT Address binding delete
- NAT port block allocation
- NAT port block release

- NAT port block active

Table 34 on page 240 describes the flow template format for NAT44 session creation and deletion events. The Information Element (IE) names and their IANA IDs are as defined in the IP Flow Information Export (IPFIX) Entities specification by the Internet Assigned Numbering Authority (IANA).

Table 34: Flow Template Format for NAT44 Session Creation and Deletion

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
sourceIPv4Address	32	8
postNATSourceIPv4Address	32	225
protocolIdentifier	8	4
sourceTransportPort	16	7
postNAPTsourceTransportPort	16	227
destinationIPv4Address	32	12
postNATDestinationIPv4Address	32	226
destinationTransportPort	16	11
postNAPTdestinationTransportPort	16	228
natOriginatingAddressRealm	8	229
natEvent	8	230
flowDurationMilliseconds	32	161

Table 34: Flow Template Format for NAT44 Session Creation and Deletion *(Continued)*

Information Element (IE)	Size (bits)	IANA ID
initiatorPackets	64	298
responderPackets	64	299
flowDirection	8	61

Table 35 on page 241 describes the flow template format for NAT64 session creation and deletion events.

Table 35: Flow Template Format for NAT64 Session Creation and Deletion

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
sourceIPv6Address	128	27
postNATSourceIPv6Address	32	225
protocolIdentifier	8	4
sourceTransportPort	16	7
postNAPTsourceTransportPort	16	227
destinationIPv6Address	128	28
postNATDestinationIPv6Address	32	226
destinationTransportPort	16	11
postNAPTdestinationTransportPort	16	228

Table 35: Flow Template Format for NAT64 Session Creation and Deletion *(Continued)*

Information Element (IE)	Size (bits)	IANA ID
natOriginatingAddressRealm	8	229
natEvent	8	230
flowDurationMilliseconds	32	161
initiatorPackets	64	298
responderPackets	64	299
flowDirection	8	61

Table 36 on page 242 describes the flow template format for NAT44 binding information base (BIB) creation and deletion events.

Table 36: Flow Template Format for NAT44 BIB Creation and Deletion

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
sourceIPv4Address	32	8
postNATSourceIPv4Address	32	225
protocolIdentifier	8	4
sourceTransportPort	16	7
postNAPTsourceTransportPort	16	227
natEvent	8	230

Table 37 on page 243 describes the flow template format for NAT64 binding information base (BIB) creation and deletion events.

Table 37: Flow Template Format for NAT64 BIB Creation and Deletion

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
sourceIPv6Address	128	27
postNATSourceIPv6Address	32	225
protocolIdentifier	8	4
sourceTransportPort	16	7
postNAPTsourceTransportPort	16	227
natEvent	8	230

Table 38 on page 243 describes the flow template format for addresses exhaustion events.

Table 38: Flow Template Format for Address Exhausted Events

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
natEvent	8	230
natPoolName	512	284

Table 39 on page 244 describes the flow template format for ports exhaustion events.

Table 39: Flow Template Format for Ports Exhausted Events

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
natEvent	8	230
postNATSourceIPv4Address	32	225
protocolIdentifier	8	4

Table 40 on page 244 describes the flow template format for NAT44 quota exceeded events.

Table 40: Flow Template Format for NAT44 Quota Exceeded Events

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
natEvent	8	230
sourceIPv4Address	32	8

Table 41 on page 244 describes the flow template format for NAT64 quota exceeded events.

Table 41: Flow Template Format for NAT64 Quota Exceeded Events

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
natEvent	8	230
sourceIPv6Address	128	27

Table 42 on page 245 describes the flow template format for NAT44 address binding creation and deletion events.

Table 42: Flow Template Format for NAT44 Address Binding Creation and Deletion Events

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
natEvent	8	230
sourceIPv4Address	32	8
postNATSourceIPv4Address	32	225

Table 43 on page 245 describes the flow template format for NAT64 address binding creation and deletion events.

Table 43: Flow Template Format for NAT64 Address Binding Creation and Deletion Events

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
natEvent	8	230
sourceIPv6Address	128	27
postNATSourceIPv4Address	32	225

Table 44 on page 246 describes the flow template format for NAT44 port block allocation and deallocation events.

Table 44: Flow Template Format for NAT44 Port Block Allocation and Deallocation Events

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
sourceIPv4Address	32	8
postNATSourceIPv4Address	32	225
portRangeStart	16	361
portRangeEnd	16	362
portRangeStepSize	16	363
portRangeNumPorts	16	364
observationTimeMilliseconds (time when PBA allocated) **NOTE**: This IE is not included in flow templates when using the MX-SPC3 services card.	64	323
natEvent	8	230

Table 45 on page 246 describes the flow template format for NAT64 port block allocation and deallocation events.

Table 45: Flow Template Format for NAT64 Port Block Allocation and Deallocation Events

Information Element (IE)	Size (bits)	IANA ID
observationTimeMilliseconds	64	323
sourceIPv6Address	128	27

Table 45: Flow Template Format for NAT64 Port Block Allocation and Deallocation Events *(Continued)*

Information Element (IE)	Size (bits)	IANA ID
postNATSourceIPv4Address	32	225
portRangeStart	16	361
portRangeEnd	16	362
portRangeStepSize	16	363
portRangeNumPorts	16	364
observationTimeMilliseconds (time when port block allocation (PBA) is configured) **NOTE**: This IE is not included in flow templates when using the MX-SPC3 services card.	64	323
natEvent	8	230

In all of the aforementioned templates, the natEvent field maps to one of the values listed in Table 46 on page 247, depending on the type of event.

Table 46: Association Between natEvent Values and Names

natEvent Value	natEvent Name
1	NAT44 Session create
2	NAT44 Session delete
3	NAT Addresses exhausted

Table 46: Association Between natEvent Values and Names *(Continued)*

natEvent Value	natEvent Name
4	NAT64 Session create
5	NAT64 Session delete
6	NAT44 BIB create
7	NAT44 BIB delete
8	NAT64 BIB create
9	NAT64 BIB delete
10	NAT ports exhausted
11	NAT Quota exceeded
12	NAT Address binding create
13	NAT Address binding delete
14	NAT port block allocation
15	NAT port block release
16	NAT port block active

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.3R2	You can also configure MX Series routers with MX-SPC3 services cards with this capability starting from Junos OS Release 19.3R2.

RELATED DOCUMENTATION

Configuring Log Generation of NAT Events in Flow Monitoring Record Format on an MX Series Router or NFX250 | 252

Monitoring NAT Events on MX Series Routers by Logging NAT Operations in Flow Template Formats | 268

Example: Configuring Logs in Flow Monitoring Format for NAT Events on MX Series Routers for Troubleshooting | 271

Configure Active Flow Monitoring Logs for NAT44/NAT64

IN THIS SECTION

Overview

Active Flow Monitoring logs are generated for NAT44 /NAT64 sessions to create or delete events on MX-SPC3 devices.

Requirements

This example uses the following hardware and software components:

- MX480 and MX960 with MX-SPC3
- Junos OS Release 21.2R1

Configuration

IN THIS SECTION

To configure Active Flow Monitoring logging on MX-SPC3 devices, perform these tasks:

1. Configure the collectors on an interface.

```
[edit]
user@host# set services jflow-log collector c1 destination-address 10.30.1.2
user@host# set services jflow-log collector c1 destination-port 1055
user@host# set services jflow-log collector c1 source-ip 10.30.1.1
```

2. Configure the collector groups.

```
[edit]
user@host# set services jflow-log collector-group cg1 collector c1
```

3. Configure the template profiles and associate the template profile with the collector or collector group.

```
[edit]
user@host# set services jflow-log template-profile t1 collector-group cg1
```

4. Associate the template profile with the template type.

```
[edit]
user@host# set services jflow-log template-profile t1 template-type nat
```

5. Associate the template profile with the version.

```
[edit]
user@host# set services jflow-log template-profile t1 version ipfix
```

6. Assign the refresh-rate values.

```
[edit]
user@host# set services jflow-log template-profile t1 refresh-rate packets 100
user@host# set services jflow-log template-profile t1 refresh-rate seconds 60
```

7. Associate the template profile with the service set.

```
[edit]
user@host# set services service-set ss1 jflow-log template-profile t1
```

Results

From the configuration mode, confirm your configuration by entering the `show services jflow-log` command in configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show services jflow-log
collector c1 {
   destination-address 10.30.1.2;
   destination-port 1055;
   source-ip 10.30.1.1;
}
collector-group cg1 {
   collector c1;
}
template-profile t1 {
   collector-group cg1;
   template-type nat;
   version ipfix;
   refresh-rate {
      packets 100;
      seconds 60;
   }
}
```

Configuring Log Generation of NAT Events in Flow Monitoring Record Format on an MX Series Router or NFX250

Keep the following points in mind when you configure the capability to generate logs or records in flow monitoring format for NAT events:

- Enabling syslog and Jflow capabilities at the same time might result in scaling impacts because both these mechanisms use a separate infrastructure to transfer records out to the collector.
- High number of NAT events can cause scalability considerations because of the flow monitoring framework too requiring system processes.
- The flow monitoring log infrastructure uses data CPUs to send the logs to the external flow server, which might cause a slight impact on performance.
- An explicit, separate maximum limit on the number of flow monitoring messages that are generated for NAT error events is implemented. You can control the maximum number of NAT error events for which logs in flow monitoring format must be recorded by including the `message-rate-limit` *`messages-per-second`* option at the `[edit interfaces` *`interface-name`* `services-options jflow-log]` hierarchy level. These records for NAT error events are generated when addresses for allocation from the NAT pool are not available, when ports for allocation to a subscriber are not available, or when the allocated quota is exceeded for NAT events (more than the configured number of ports is requested). Also, you can configure the `message-rate-limit` option that previously existed at the `[edit interfaces` *`interface-name`* `services-options syslog]` hierarchy level to specify the maximum number of system log messages per second that can be sent from the PIC to either the Routing Engine (local) or to an external server (remote).
- NAT error events such as “Out of Ports”, “Out of Addresses” and “Quota Exceeded” are rate limited. Default rate limit is 10,000 events per second. This setting is also configurable at PIC level.
- The template for NAT event logging is in accordance with IETF as *IPFIX Information Elements for logging NAT Events—draft-ietf-behave-ipfix-nat-logging-02*.
- Only UDP-based logging is supported, which is an unreliable protocol.
- This functionality is supported on MX Series routers with Junos OS Extension-Provider packages installed and configured on the device, and on MS-MPCs, MS-PICs, and MX-SPC3s. It is not supported on MS-DPCs with MX Series routers.
- Transmission of logs occurs in clear-text format similar to other log messages that the services PICs do not encrypt. It is assumed that the transport of logs and the positioning of the log collector are within a secured realm. Because the messages do not contain sensitive details such as username or passwords, the messages do not cause any security or reliability risks.

- Template IDs 0 through 255 are reserved for template sets and the maximum number of templates supported for logging events in flow monitoring format is 255. When you modify a template-profile configuration (changes to the collector or version, or a deactivation and activation of the service set associated with the template), the specific template is deregistered and reregistered. However, the flow monitoring infrastructure requires 10 minutes by default as the delay period for the template IDs to be freed up. As a result, if you modify the template-profile settings many times within the 10-minute period, the maximum limit on the template IDs of 255 is exceeded and further templates are not registered.
  In such a case, when templates are not being registered, you must wait until the delay period for deleting a deregistered template of 10 minutes before you perform any more configuration changes to have templates registered with the flow monitoring application. To examine whether a template has been registered, you can use the `show services service-sets statistics jflow-log` command. If the Sent field displays a non-zero value for template records, it denotes that templates are successfully registered.

- In a scenario in which the capability to log NAT events in flow monitoring format is enabled at the service-set level, and if the PIC boots up, the flow monitoring log templates are registered with the flow monitoring application. During the registration process, a first set of 12 template records are sent to the collector. However, all of the template records might not reach the collector from the PIC on the router or might not be transmitted out of the router because the interface might not be up from the perspective of the Packet Forwarding Engine. After the refresh time of a template expires, next set of template records are sent out to the collector. For example, if the template refresh time is 60 seconds, only after 60 seconds from the time of booting of the PIC, template records are properly sent to the collector.

- If no problems occur in the transmission of flow monitoring log messages to the collector from the PIC, the Sent field is incremented to indicate NAT events being logged for every event. Also, the tcpdump utility at the destination IP address of the collector denotes the reception of UDP packets. If NAT processing occurs and the value in the `Dropped` section of the output of the `show services service-sets statistics jflow-log service-set` *`service-set-name`* command is incremented or not incremented, you must examine the debugging statistics and counters to determine if any problems exist in the network for transmission of the flow monitoring log messages.

RELATED DOCUMENTATION

Understanding NAT Event Logging in Flow Monitoring Format on an MX Series Router or NFX250 | 236

Monitoring NAT Events on MX Series Routers by Logging NAT Operations in Flow Template Formats | 268

Example: Configuring Logs in Flow Monitoring Format for NAT Events on MX Series Routers for Troubleshooting | 271

Exporting Syslog Messages to an External Host Without Flow Monitoring Formats Using an MX Series Router or NFX250

Until Junos OS Release 14.2R1, the only mechanism you can use to generate logs for NAT sessions was by enabling system logging for service sets and transferring syslog messages to either the internal local host on the Routing Engine or to an external host server. When a syslog is enabled with the class or component being NAT logs and session logs configured, NAT events are recorded. A sample of one such syslog output is as follows:

```
{service_set_3}[jservices-nat]: JSERVICES_NAT_RULE_MATCH: proto 17(UDP) app: any, xe-3/1/1.0#012
192.0.2.2/18575 -> 23.0.0.2/63,Match NAT rule-set (null) rule nat-basic_1
term t1
{service_set_3}MSVCS_LOG_SESSION_OPEN: App:none, xe-3/1/1.0#012 24.0.0.2:18575
[198.51.100.17:1048] -> 23.0.0.2:63  (UDP)
{service_set_3}MSVCS_LOG_SESSION_CLOSE: App:none, xe-3/1/1.0#012 24.0.0.2:18575
[198.51.100.17:1048] -> 23.0.0.2:63  (UDP)
```

From the preceding syslog output, it denotes that NAT create log (NAT translation) and delete log (NAT release) are generated during session events as a part of session-logs configuration. Another important log that is NAT pool exhaustion (not illustrated in the preceding example) is generated as a part of NAT-logs configuration. Such an event message might be caused by Address pooling paired (APP), endpoint-independent mapping (EIM), or address and port exhaustion.

Exporting Version 9 Flow Data Records to a Log Collector Overview Using an MX Series Router or NFX250

A flow record template defines a collection of fields with corresponding descriptions of the format and syntax for the elements or attributes that are contained in it. Network elements (such as routers and switches), which are called exports, accumulate the flow data and export the information to collectors, which are hosts or external devices that can save a large volume of such system log messages for events or system operations. The collected data provides granular, finer-level metering and statistical data for highly flexible and detailed resource usage accounting. Templates that are sent to the collector contain the structural information about the exported flow record fields; therefore, if the collector cannot interpret the formats of the new fields, it can still process the flow record.

The version 9 flow template has a predefined format. An export packet consists of a packet header followed by one or more FlowSet fields. The FlowSet fields can be any of the possible three types—Template, Data, or Options Template. The template flowset describes the fields that are in the data flowsets (or flow records). Each data flowset contains the values or statistics of one or more flows with the same template ID. An interleaved NetFlow version 9 export packet contains the packet header, Template FlowSet, and Data FlowSet fields. A Template FlowSet field signifies each event such as the creation of a NAT entry or the release of a NAT entry allocated, and the Data FlowSet field denotes the NAT sessions for which the Template FlowSet (or the event type) is associated. For example, if a NAT address entry creation, exhaustion of addresses in a NAT pool, and a NAT entry deletion or release occur, an interleaved version 9 export packet contains the packet header, one Template FlowSet field for NAT address creation, two Data FlowSet fields for the two sessions for which address creation is performed, another TemplateSet field for NAT address deletion, two Data FlowSet fields for the two sessions for which address deletion event occurs, and the other TemplateSet field for NAT pool consumption having exceeded the configured number of pools.

The following are the possible combinations that can occur in an export packet:

- An export packet that consists of interleaved template and data FlowSets—A collector device should not assume that the template IDs defined in such a packet have any specific relationship to the data FlowSets within the same packet. The collector must always cache any received templates, and examine the template cache to determine the appropriate template ID to interpret a data record.

- An export packet consisting entirely of data FlowSets—After the appropriate template IDs have been defined and transmitted to the collector device, most of the export packets consist solely of data FlowSets.

- An export packet consisting entirely of template FlowSets—Although this case is the exception, it is possible to receive packets containing only template records. Ordinarily, templates are appended to data FlowSets. However, in some instances only templates are sent. When a router first boots up or reboots, it attempts to synchronize with the collector device as quickly as possible. The router can send template FlowSets at an accelerated rate so that the collector device has sufficient information to parse any subsequent data FlowSets. Also, template records have a limited lifetime, and they must be periodically refreshed. If the refresh interval for a template occurs and no appropriate data FlowSet that needs to be sent to the collector device is present, an export packet consisting only of template FlowSets is sent.

Understanding Exporting IPFIX Flow Data Records to a Log Collector Using an MX Series Router or NFX250

The IPFIX protocol enables you to access IP flow information on MX Series Routers or an NFX250 device. The IPFIX collection process receives the flow information traversing through multiple network elements within the data network in a consistent, identical manner of representation and communication of traffic flows from the network elements to the collection point. An IPFIX device hosts at least one exporting process, which transmits flow records to collecting processes. A collector is a device that performs the collecting processes and an exporter is a device that performs the transfer to data to a collector. An IPFIX message consists of a message header followed by one or more Sets. The Sets can be any of the possible three types: Data Set, Template Set, or Options Template Set. Flow monitoring version 10 (IPFIX) message formats are very similar to version 9 message patterns.

The message header contains the following fields:

- Version—Version of the flow record format exported in this message. The value of this field is 0x000a.
- Length—Total length of the IPFIX message, measured in octets, including the header and Sets fields.
- Export Time—Time, in seconds, since midnight Coordinated Universal Time (UTC) of January 1, 1970, at which the IPFIX message header leaves the exporter.
- Sequence Number—Incremental sequence counter with a value of 2^32 (2 raised to the power of 32) of all IPFIX data records sent from the current Observation Domain by the exporting process. Template and Options Template records do not increase the Sequence Number attribute.
- Observation Domain ID—A 32-bit identifier of the Observation Domain that is locally unique to the exporter.

One of the essential elements in the IPFIX record format is the Template Flow Set record. Templates vastly enhance the flexibility of the Flow Record format because they allow the collector to process Flow Records without necessarily knowing the interpretation of all the data in the Flow Record. A Template Record contains any combination of Internet Assigned Numbers Authority (IANA)-assigned and/or enterprise-specific information element identifiers.

The format of the Template Record signifies a template record header and one or more Field Specifier attributes. The Template Flow Set record contains the following fields:

- Enterprise bit—This is the first bit of the Field Specifier. If this bit is zero, the Information Element Identifier identifies an IETF-specified Information Element, and the four-octet Enterprise Number field must not be present. If this bit is one, the Information Element identifier identifies an enterprise-specific Information Element, and the Enterprise Number field must be present.
- Information Element identifier—An Information Element is a protocol and encoding-independent description of an attribute that can appear in an IPFIX Record. It is a numeric value that represents the type of Information Element.
- Field Length—Length of the corresponding encoded Information Element, in octets. The value 65535 is reserved for variable-length Information Elements.
- Enterprise Number—IANA enterprise number of the authority defining the Information Element identifier in this Template Record.

The Data Records are sent in Data Sets. The Data Record field consists only of a Set Header and one or more Field Values. The Template ID to which the Field Values belong is encoded in the Set Header field

"Set ID" ("Set ID" = "Template ID"). Interpretation of the Data Record format can be done only if the Template Record corresponding to the Template ID is available at the collecting procedure. Field Values do not necessarily have a length of 16 bits and are encoded according to their data type specified.

Mapping Between Field Values for Version 9 Flow Templates and Logs Exported From an MX-Series Router or NFX250

The following table describes different field IDs or values for flow monitoring logs generated for NAT events in version 9 flow record formats and the events that correspond to the field values:

Field ID	Name	Size (Bytes)	Description
8	ipv4 src address	4	IPv4 source address
225	natInsideGlobalAddress	4	It reports a modified value caused by a NAT middlebox (forwarding class and loss priority) represents function after the packet passed the Observation Point.
12	ipv4 destination address	4	IPv4 destination address
226	natOutsideGlobalAddress	4	It reports a modified value caused by a NAT middlebox function after the packet passed the Observation Point.
7	transport source-port	2	TCP/UDP source port
227	postNAPTSourceTransportPort	2	It reports a modified value caused by a Network Address Port Translation (NAPT) middlebox function after the packet passed the Observation Point.
11	transport destination-port	2	TCP/UDP destination port

(Continued)

Field ID	Name	Size (Bytes)	Description
228	postNAPTDestinationTransportPort	2	It reports a modified value caused by a Network Address Port Translation (NAPT) middlebox function after the packet passed the Observation Point.
234	ingressVRFID	4	Unique identifier of the VRF name where the packets of this flow are being received. This identifier is unique per Metering Process.
235	egressVRFID	4	Unique identifier of the VRF name where the packets of this flow are being sent. This identifier is unique per Metering Process.
4	Ip protocol	1	IP protocol byte
229	natOriginatingAddressRealm	1	Indicates whether the session was created because traffic originated in the private or public address realm. postNATSourceIPv4Address, postNATDestinationIPv4Address, postNAPTSourceTransportPort, and postNAPTDestinationTransportPort are qualified with the address realm in perspective. The allowed values are: Private: 1 Public: 2
230	natEvent	1	Indicates a NAT event. The allowed values are: 1 - Create event. 2 - Delete event. 3 - Pool exhausted. A Create event is generated when a NAT translation is created, whether dynamically or statically. A Delete event is generated when a NAT translation is deleted.
1	inBytes	N	Incoming counter with length N x 8 bits for the number of bytes associated with an IP Flow. By default N is 4

(Continued)

Field ID	Name	Size (Bytes)	Description
2	inPkts	N	Incoming counter with length N x 8 bits for the number of packets associated with an IP Flow. By default N is 4
323	observationTimeMilliseconds	8	Specifies the absolute time in milliseconds of an observation that represents a time value in units of milliseconds based on coordinated universal time (UTC). The choice of an epoch, for example, 00:00 UTC, January 1, 1970, is left to corresponding encoding specifications for this type. Leap seconds are excluded. Note that transformation of values might be required between different encodings if different epoch values are used.
27	sourceIPv6Address	16	IPv6 source address
284	natPoolName	64	NAT resource pool name
361	portRangeStart	2	The port number identifying the start of a range of ports. A value of zero indicates that the range start is not specified, ie the range is defined in some other way.
362	portRangeEnd	2	The port number identifying the end of a range of ports. A value of zero indicates that the range end is not specified, and the range is defined in some other way.
363	portRangeStepSize	2	The step size in a port range. The default step size is 1, which indicates contiguous ports. A value of zero indicates that the step size is not specified, and the range is defined in some other way.

(Continued)

Field ID	Name	Size (Bytes)	Description
364	portRangeNumPorts	2	The number of ports in a port range. A value of zero indicates that the number of ports is not specified, and the range is defined in some other way.

Consider a sample scenario of a NAT address creation event. Based on the fields in the preceding table, for translations that are not available (such as natOutsideGlobalAddress) is set to 0. Ingress and Egress VRF of the flow can be made available. Also, natEvent is equal to 1 (create). The inBytes field is assumed to be 0 or number of bytes of the incoming packet and the inPkts field is either 0 or 1 because it is the first packet into the system when translation happens. The observationTimeMilliseconds field denotes the time when this address translation creation is recorded.

For a NAT address deletion event, for translations that are not available (such as natOutsideGlobalAddress) is set to 0. Ingress and Egress VRF of the flow can be made available. Also, natEvent is equal to 2 (create). The inBytes field denotes the number of bytes for this flow in both the forward or upward, the value of the inPkts field denotes the number of packets for this flow in both the upward and backward directions. observationTimeMilliseconds is the time when this deletion of translation is recorded.

When the NAT pool is exhausted and no further addresses are remaining for allocation, for translations that are not available (such as natOutsideGlobalAddress) is set to 0. Ingress and Egress VRF of the flow can be made available. Also, the natEvent field is set to 3 (Pool exhausted). All resource failures are combined as a single event. The inBytes field is assumed to be 0 or number of bytes of the incoming packet and the inPkts field is either 0 or 1 because it is the first packet into the system when translation happens. The value of the observationTimeMilliseconds field is the time when this failed translation is recorded.

Mapping Between Field Values for IPFIX Flow Templates and Logs Exported From an MX Series Router or NFX250

An IETF draft defining IPFIX Information Elements for logging various NAT events is available in IETF as *IPFIX Information Elements for logging NAT Events—draft-ietf-behave-ipfix-nat-logging-02*. The flow monitoring template format for flow monitoring logs generated for NAT events comply with the templates defined in this draft for logging NAT44/NAT64 session create/delete, binding information base (BIB) create/delete, address exhaust, pool exhaustion, quota exceeded, address binding create/delete, port block allocation and de-allocation events. Also, this draft has an extension for NAT64.

Support is implemented for logging events for both NAT44 and NAT64. Apart from those templates defined in this draft, no new user-defined templates are created for logging any NAT events.

The following table lists the extensions to the NAT events. The data record contains the corresponding natEvent value to identify the event that is being logged.

Event Name	Values
NAT44 Session create	1
NAT44 Session delete	2
NAT Addresses exhausted	3
NAT64 Session create	4
NAT64 Session delete	5
NAT44 BIB create	6
NAT44 BIB delete	7
NAT64 BIB create	8
NAT64 BIB delete	9
NAT ports exhausted	10
Quota exceeded	11
Address binding create	12
Address binding delete	13
Port block allocation	14

(Continued)

Event Name	Values
Port block deallocation	15

The following table describes the field IDs or values and the corresponding names for IPv6 addresses for IPFIX flows:

Field ID	Name	Size (Bytes)	Description
27	sourceIPv6Address	16	IPv6 source address
28	destinationIPv6Address	16	IPv6 destination address
281	postNATSourceIPv6Address	16	Translated source IPv6 address
282	postNATDestinationPv6Address	16	Translated destination IPv6 address

The following table describes the field names and whether they are required or not for NAT64 session creation and deletion events:

Field Name	Size (Bits)	Whether the Field Is Mandatory
timeStamp	64	Yes
vlanID/ingressVRFID	32	No
sourceIPv4Address	128	Yes
postNATSourceIPv4Address	32	Yes
protocolIdentifier	8	Yes
sourceTransportPort	16	Yes

(Continued)

Field Name	Size (Bits)	Whether the Field Is Mandatory
postNAPTsourceTransportPort	16	Yes
destinationIPv4Address	128	No
postNATDestinationIPv4Address	32	No
destinationTransportPort	16	No
postNAPTdestinationTransportPort	16	No
natOriginatingAddressRealm	8	No
initiatorOctets	64	No
responderOctets	64	No
flowEndReason	8	No
natEvent	8	Yes

A NAT44 session creation template record can contain the following fields. The natEvent field contains a value of 1, which indicates a NAT44 session creation event. An example of such a template is as follows:

Field Name	Size (Bits)	Value
timeStamp	64	09:20:10:789
sourceIPv4Address	32	192.168.16.1
postNATSourceIPv4Address	32	192.0.2.100

(Continued)

Field Name	Size (Bits)	Value
protocolIdentifier	8	TC
sourceTransportPort	16	14800
postNAPTsourceTransportPort	16	1024
destinationIPv4Address	32	198.51.100.104
postNATDestinationIPv4Address	32	198.51.100.104
destinationTransportPort	16	80
postNAPTdestinationTransportPort	16	80
natOriginatingAddressRealm	8	0
initiatorOctets	64	No
responderOctets	64	No
flowEndReason	8	No
natEvent	8	1

A NAT44 session deletion template record can contain the following fields. The natEvent field contains a value of 2, which indicates a NAT44 session deletion event. An example of such a template is as follows:

Field Name	Size (Bits)	Value
timeStamp	64	09:20:10:789

(Continued)

Field Name	Size (Bits)	Value
sourceIPv4Address	32	192.168.16.1
postNATSourceIPv4Address	32	192.0.2.100
protocolIdentifier	8	TC
sourceTransportPort	16	14800
postNAPTsourceTransportPort	16	1024
destinationIPv4Address	32	198.51.100.104
postNATDestinationIPv4Address	32	198.51.100.104
destinationTransportPort	16	80
postNAPTdestinationTransportPort	16	80
natOriginatingAddressRealm	8	0
natEvent	8	2

To support all session termination reasons on NAT, existing `flowEndReason` information element is extended. A new CLI command `session-end-reason` is introduced to configure `flowEndReason` to be a part of J-Flow IPFIX template.

If the CLI is not configured or configured as default, the `flowEndReason` exports the default set information to fill in the data records. If the CLI is configured as custom, the `flowEndReason` exports the custom set information to fill in the data records.

The table lists the set of session termination values that can be exported:

Table 47: Session Termination Values

Session Close Reason	Session Close Reason string	Scenarios/Remark	Custom Set values	Default Set values
NAT_SESSION_CREATION	idle Timeout	When any session gets timeout	0x01	0x01
NAT_SESSION_CLOSE_TCP_CLIENT_RST	TCP CLIENT RST	Receives a TCP packet from Client with RST FLAG set	0x13	0xFF
NAT_SESSION_CLOSE_TCP_SERVER_RST	TCP SERVER RST	Receives a TCP packet from Server with RST FLAG set	0x23	0xFF
NAT_SESSION_CLOSE_TCP_FIN	TCP FIN	Receives FIN Packet	0x03	0x03
NAT_SESSION_CLOSE_ICMP_ERR	ICMP Error	Receiving ICMP Error packet in Fast path. icmp related error messages mentioned below	0x10	0XFF

Table 47: Session Termination Values *(Continued)*

Session Close Reason	Session Close Reason string	Scenarios/Remark	Custom Set values	Default Set values
NAT_SESSION_CLOSE_NSRP	HA	Create a NAT session on active router. Now, Switch to backup Router Manually or by bringing down the pic on active router. Wait for the switchover and send traffic. Ensure the session is synchronized. Now close the session.	0x20	0xFF
NAT_SESSION_CLOSE_POLICY_DELETE	policy delete	When you delete Policy rematch configuration with active session.	0x50	0xFF
NAT_SESSION_CLOSE_POLICY_UPDATE	policy update	When you Update Policy rematch configuration with active session.	0x60	0xFF
NAT_SESSION_CLOSE_JSF_PLUGIN	application failure or action	It is a very rare scenario and would be difficult to simulate. Please don't have test case for this.	0x70	0xFF

Table 47: Session Termination Values *(Continued)*

Session Close Reason	Session Close Reason string	Scenarios/Remark	Custom Set values	Default Set values
NAT_SESSION_CLOSE_IFP_ZONECHANGED_SSCAN	session interface zone changed	when redundancy switchover happens in ams interface	0x80	0xFF
NAT_SESSION_CLOSE_CLI	CLI	Force clear the session	0x04	0x04

Monitoring NAT Events on MX Series Routers by Logging NAT Operations in Flow Template Formats

You can configure MX Series routers with MS-MPCs, MS-MICs, and MX-SPC3s to log network address translation (NAT) events using the Junos Traffic Vision (previously known as Jflow) version 9 or IPFIX (version 10) template format. NAT event logger generates logs or template records in flow monitoring format and transmits them to the specified external collector or server for various NAT events, such as NAT44 and NAT64 session creation and deletion, and NAT44 and NAT64 binding information base events.

NOTE: This functionality is supported on MX Series routers with Junos OS Extension-Provider packages installed and configured on the device, and on MS-MPCs, MS-PICs, and MX-SPC3s. It is not supported on MS-DPCs with MX Series routers.

You can configure the mechanism to record logging messages in flow monitoring format for NAT events. You need to define collectors, and template profiles that contain the properties for flow monitoring logs. You can create a template profile for a particular NAT service on an MX Series router with MS-MPCs, MS-MICs, or MX-SPC3s, or for a service set, which applies for all of the NAT services. You can define a template profile to generate flow monitoring logs in a specific flow template format and associate the specified template profile with a service set.

To enable the flow monitoring log capability for NAT events and configure the transmission of logs to collectors at a service level:

1. Define the flow monitoring log service to be applied on an interface to control the maximum number of flow monitoring logs generated for NAT error events.

```
[edit]
user@host# set interfaces ms-fpc/pic/port services-options jflow-log message-rate-limit
messages-per-second
```

For example:

```
[edit]
user@host# set interfaces ms-5/0/0 services-options jflow-log message-rate-limit 50
```

2. Configure the collectors and collector groups.

```
[edit]
user@host# set services jflow-log collector collector-name destination-address address
destination-port port-number source-ip address
user@host# set services jflow-log collector-group collector-group-name collector [ collector-
name1 collector-name2]
```

For example:

```
[edit]
user@host# set services jflow-log collector c1 destination-address 203.0.113.3 destination-
port 1 source-ip 192.0.2.1
user@host# set services jflow-log collector-group cg1 collector c1
```

3. Configure the template profiles and associate the template profile with the collector or collector group.

```
[edit]
user@host# set services jflow-log template-profile template-profile-name collector collector-
name version (ipfix | v9) template-type nat refresh-rate packets packets seconds seconds
user@host# set services jflow-log template-profile template-profile-name collector-group
collector-group-name version (ipfix | v9) template-type nat refresh-rate packets packets
seconds seconds
```

For example:

```
[edit]
user@host# set services jflow-log template-profile t1 collector c1 version ipfix template-
type nat refresh-rate packets 20 seconds 20
user@host# set services jflow-log template-profile t1 collector-group cg1
user@host# set services jflow-log template-profile t2 collector c2 version v9 template-type
nat refresh-rate packets 20 seconds 20
```

4. Associate the template profile with the service set.

```
[edit]
user@host# set services service-set service-set-name jflow-log template-profile template-
profile-name
```

For example:

```
[edit]
user@host# set services service-set sset_0 jflow-log template-profile t1
```

RELATED DOCUMENTATION

Understanding NAT Event Logging in Flow Monitoring Format on an MX Series Router or NFX250 | 236

Configuring Log Generation of NAT Events in Flow Monitoring Record Format on an MX Series Router or NFX250 | 252

Example: Configuring Logs in Flow Monitoring Format for NAT Events on MX Series Routers for Troubleshooting | 271

Example: Configuring Logs in Flow Monitoring Format for NAT Events on MX Series Routers for Troubleshooting

IN THIS SECTION

You can configure MX Series routers with MS-MPCs, MS-MICs, and MX-SPC3s to log network address translation (NAT) events using the Junos Traffic Vision (previously known as Jflow) version 9 or IPFIX (version 10) template format. This method of generating flow monitoring records for NAT events, such as NAT44 and NAT64 session creation and deletion, and NAT44 and NAT64 binding information base events, enables cohesive and streamlined analysis of NAT traffic and troubleshooting of NAT-related problems.

> **NOTE**: This functionality is supported on MX Series routers with Junos OS Extension-Provider packages installed and configured on the device, and on MS-MPCs, MS-PICs, and MX-SPC3s. It is not supported on MS-DPCs with MX Series routers.

This example describes how to configure flow monitoring log generation in flow monitoring format for NAT events at the service-set level on MS-MIC, MS-MPC, and MX-SPC3, and contains the following sections:

> **NOTE**: This configuration example is for an Interface-Style service set.

Requirements

This example uses the following hardware and software components:

- One MX Series router with an MS-MPC, MS-MIC, or MX-SPC3
- Junos OS Release 14.2R2 or later for MX Series routers

Generation of Log Messages Using Flow Templates for NAT Operations on MS-MPCs, MS-MICs, and MX-SPC3s

You can configure the mechanism to record logging messages in flow monitoring format for NAT events. You can create a template profile for a particular NAT service on an MX Series router with MS-MPCs, MS-MICs, or MX-SPC3s, or for a service set, which applies for all of the NAT services. You must define a template profile to generate flow monitoring logs in a specific flow template format and attach the template profile with a service set. You must configure a collector or a group of collectors, which are hosts that receive the log messages for NAT events from the service PIC or the exporter. You need to associate a template profile with the collector. The profile defines the characteristics of the flow monitoring record template, such as the version of flow monitoring (version 9 or IPFIX), the refresh rate, in either packets or seconds, and the type of service or application (NAT in this case) for which flow records must be sent to the collector.

Assume a sample deployment in which two collectors, c1 and c2, are defined. These collectors are clustered into two groups. The collector group, cg1, contains c1 and c2, and the collector group, cg2, contains c2. Two template profiles named t1 and t2 are defined. The profiles, t1 and t2, are associated with collectors, c1 and c2, respectively.

These profiles describe the properties or attributes for transmission of logs, such as the flow template format to be used, the rate at which the logs must be refreshed, and the service or event, such as NAT, for which logs must be sent to the specified collector.

Configuration

IN THIS SECTION

To enable the flow monitoring log capability for NAT events and configure the transmission of logs to collectors, perform these tasks:

CLI Quick Configuration

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

Configuring Service Set Properties

```
set services service-set sset_0 interface-service service-interface ms-5/0/0.0
```

Applying Flow Monitoring Log Service on an Interface

```
set interfaces ms-5/0/0 services-options jflow-log message-rate-limit 50000
```

Enabling and Configuring Flow Monitoring Logs for a Service Set

```
set services jflow-log collector c1 destination-address 192.0.2.3 destination-port 1 source-ip
198.51.100.1
set services jflow-log collector c2 destination-address 203.0.113.5 destination-port 3 source-ip
198.51.100.2
set services jflow-log collector-group cg1 collector [ c1 c2 ]
set services jflow-log template-profile t1 collector c1 version ipfix template-type nat refresh-
rate packets 20 seconds 20
set services jflow-log template-profile t2 collector c2 version v9 template-type nat refresh-
rate packets 20 seconds 20
set services jflow-log template-profile t1 collector-group cg1
```

Associating the Template Profile with a Service Set

```
set services service-set sset_0 jflow-log template-profile t1
```

Procedure

Step-by-Step Procedure

To configure the generation and transmission of flow monitoring template logs for NAT events:

1. Create a service set properties.

   ```
   [edit]
   user@host# set services service-set sset_0 interface-service service-interface ms-5/0/0.0
   ```

2. Define the flow monitoring log service to be applied on an interface.

```
[edit]
user@host# set interfaces ms-5/0/0 services-options jflow-log message-rate-limit 50000
```

3. Configure the collectors and collector groups.

```
[edit]
user@host# set services jflow-log collector c1 destination-address 192.0.2.3 destination-port
1 source-ip 198.51.100.1
user@host# set services jflow-log collector c2 destination-address 203.0.113.5 destination-
port 3 source-ip 198.51.100.2
user@host# set services jflow-log collector-group cg1 collector [ c1 c2 ]
user@host# set services jflow-log collector-group cg2 collector c2
```

4. Configure the template profiles and associate the template profile with the collector.

```
[edit]
user@host# set services jflow-log template-profile t1 collector c1 version ipfix template-
type nat refresh-rate packets 20 seconds 20
user@host# set services jflow-log template-profile t2 collector c2 version v9 template-type
nat refresh-rate packets 20 seconds 20
```

5. Associate the template profile with the service set.

```
[edit]
user @ host# set services service-set sset_0 jflow-log template-profile t1
```

Results

From the configuration mode, confirm your configuration by entering the `show services`, `show services jflow-log`, and `show services service-set sset_0 jflow-log` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show services
service-set sset_0 {
   interface-service {
      service-interface ms-5/0/0;
```

```
   }
}
[edit interfaces]
ms-5/0/0 {
  services-options {
   jflow-log {
    message-rate-limit 50000;
   }
  }
}

user@host# show services jflow-log
collector c1 {
    destination-address 192.0.2.3;
    destination-port 1;
    source-ip 198.51.100.1;
}
collector c2 {
    destination-address 203.0.113.5;
    destination-port 3;
    source-ip 198.51.100.2;
}
collector-group cg1 {
    collector [ c2 c1 ];
}
collector-group cg2 {
    collector c2;
}
template-profile t2 {
    collector c2;
    template-type nat;
    referesh-rate packets 20 seconds 20;
    version v9;
}
template-profile t1 {
    collector c1;
    template-type nat;
    referesh-rate packets 20 seconds 20;
    version ipfix;
}


[edit]
```

```
user@host# show services service-set sset_0 jflow-log
template-profile t2;
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform the following:

Verifying That the Flow Monitoring Logs Are Generated and Sent to Collectors

Purpose

Verify that the flow monitoring log messages in the defined template format, such as IPFIX or version 9, are generated and transmitted to the configured collectors for the different NAT operations.

Action

From operational mode, use the `show services service-sets statistics jflow-log` command:

```
user@host> show services service-sets statistics jflow-log
Interface: ms-5/0/0
  Rate limit: 1000
  Template records:
    Sent: 36
    Dropped: 0
  Data records:
    Sent: 2
    Dropped: 0

    Service-set: sset_0
      Unresolvable collectors: 0
      Template records:
        Sent: 36
        Dropped: 0
      Data records:
```

```
        Sent: 2
        Dropped: 0
```

From operational mode, use the `show services service-sets statistics jflow-log detail` command:

```
user@host> show services service-sets statistics jflow-log detail

Interface: ms-5/0/0
  Rate limit: 1000
  Template records:
    Sent: 48
    Dropped: 0
  Data records:
    Sent: 4
    Dropped: 0

    Service-set: sset_0
      Unresolvable collectors: 0
      Template records:
        Sent: 48
        Dropped: 0
      Data records:
        Sent: 4
        Dropped: 0
      NAT44 Session logs:
        Template records:
          Sent: 4
          Dropped: 0 (socket send error: 0, no memory: 0)
        Data records:
          Sent: 4
          Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
      NAT64 Session logs:
        Template records:
          Sent: 4
          Dropped: 0 (socket send error: 0, no memory: 0)
        Data records:
          Sent: 0
          Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
      NAT44 BIB logs:
```

```
    Template records:
      Sent: 4
      Dropped: 0 (socket send error: 0, no memory: 0)
    Data records:
      Sent: 0
      Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
  NAT64 BIB logs:
    Template records:
      Sent: 4
      Dropped: 0 (socket send error: 0, no memory: 0)
    Data records:
      Sent: 0
      Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
  NAT Address Exhausted logs:
    Template records:
      Sent: 4
      Dropped: 0 (socket send error: 0, no memory: 0)
    Data records:
      Sent: 0
      Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
  NAT Port Exhausted logs:
    Template records:
      Sent: 4
      Dropped: 0 (socket send error: 0, no memory: 0)
    Data records:
      Sent: 0
      Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
  NAT44 Quota Exceeded logs:
    Template records:
      Sent: 4
      Dropped: 0 (socket send error: 0, no memory: 0)
    Data records:
      Sent: 0
      Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
  NAT64 Quota Exceeded logs:
    Template records:
      Sent: 4
      Dropped: 0 (socket send error: 0, no memory: 0)
    Data records:
      Sent: 0
      Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
  NAT44 Address Bind logs:
    Template records:
```

```
      Sent: 4
      Dropped: 0 (socket send error: 0, no memory: 0)
    Data records:
      Sent: 0
      Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
  NAT64 Address Bind logs:
    Template records:
      Sent: 4
      Dropped: 0 (socket send error: 0, no memory: 0)
    Data records:
      Sent: 0
      Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
  NAT44 PBA logs:
    Template records:
      Sent: 4
      Dropped: 0 (socket send error: 0, no memory: 0)
    Data records:
      Sent: 0
      Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
  NAT64 PBA logs:
    Template records:
      Sent: 4
      Dropped: 0 (socket send error: 0, no memory: 0)
    Data records:
      Sent: 0
      Dropped: 0 (invalid data: 0, no memory: 0, above rate limit: 0)
```

Meaning

The output shows that the log messages in flow monitoring format associated with the specified service set and interface are generated for the different NAT events.

RELATED DOCUMENTATION

Understanding NAT Event Logging in Flow Monitoring Format on an MX Series Router or NFX250 | 236

Configuring Log Generation of NAT Events in Flow Monitoring Record Format on an MX Series Router or NFX250 | 252

Monitoring NAT Events on MX Series Routers by Logging NAT Operations in Flow Template Formats | 268

2

PART

Flow Capture Services

CHAPTER 6

Dynamically Capturing Packet Flows Using Junos Capture Vision

IN THIS CHAPTER

Understanding Dynamic Flow Capture

IN THIS SECTION

Dynamic flow capture enables you to capture packet flows on the basis of dynamic filtering criteria. Specifically, you can use this feature to forward passively monitored packet flows that match a particular filter list to one or more destinations using an on-demand control protocol.

Dynamic Flow Capture Architecture

The architecture consists of one or more *control sources* that send requests to a Juniper Networks router to monitor incoming data, and then forward any packets that match specific filter criteria to a set of one or more *content destinations*. The architectural components are defined as follows:

- Control source—A client that monitors electronic data or voice transfer over the network. The control source sends filter requests to the Juniper Networks router using the Dynamic Task Control Protocol (DTCP), specified in draft-cavuto-dtcp-03.txt at http://www.ietf.org/internet-drafts. The control source is identified by a unique identifier and an optional list of IP addresses.

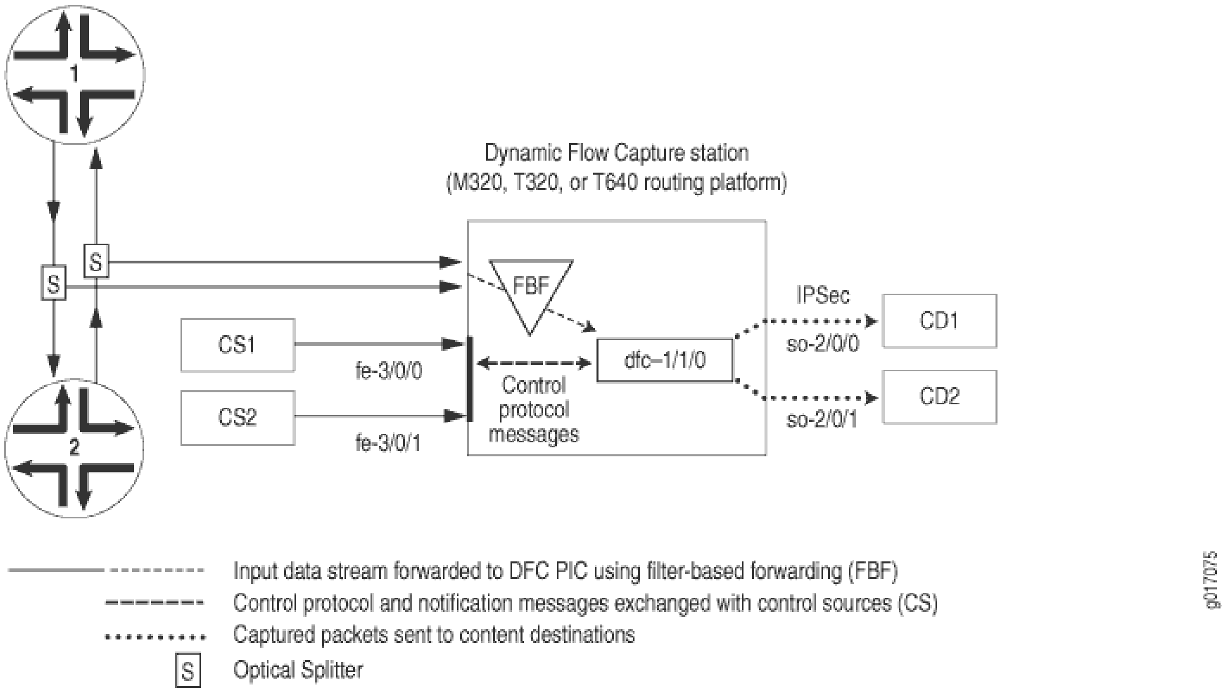
- Monitoring platform—An MX Series router containing one or more Dynamic Flow Capture (DFC) PICs, which support dynamic flow capture processing. The monitoring platform processes the requests from the control sources, creates the filters, monitors incoming data flows, and sends the matched packets to the appropriate content destinations.

- Content destination—Recipient of the matched packets from the monitoring platform. Typically the matched packets are sent using an IP Security (IPsec) tunnel from the monitoring platform to another router connected to the content destination. The content destination and the control source can be physically located on the same host. For more information on IPsec tunnels, see *Understanding Junos VPN Site Secure*.

NOTE: The dynamic flow capture PIC (either a Monitoring Services III PIC or Multiservices 400 PIC) forwards the entire packet content to the content destination, rather than to a content record as is done with cflowd or flow aggregation version 9 templates.

Figure 25 on page 283 shows a sample topology. The number of control sources and content destinations is arbitrary.

Figure 25: Dynamic Flow Capture Topology



Liberal Sequence Windowing

Each DTCP packet (add, delete, list, and refresh packets) contains a 64-bit sequence number to identify the order of the packets. Because the network is connectionless, the DTCP packets can arrive out of order to the router running the dynamic flow capture application.

The *liberal sequence window* feature implements a negative window for the sequence numbers received in the DTCP packets. It enables the dynamic flow capture application to accept not only DTCP packets with sequence numbers greater than those previously received, but also DTCP packets with lesser sequence numbers, up to a certain limit. This limit is the negative window size; the positive and negative window sizes are +256 and –256 respectively, relative to the current maximum sequence number received. No configuration is required to activate this feature; the window sizes are hard-coded and nonconfigurable.

Intercepting IPv6 Flows

Dynamic flow capture can intercept passively monitored IPv6 traffic only. All support for IPv4 interception remains the same. The interception of IPv6 traffic happens in the same way the filters capture IPv4 flows. With the introduction of IPv6 interception, both IPv4 and IPv6 filters can coexist. The mediation device, however, cannot be located in an IPv6 network.

Dynamic flow capture does not support interception of VPLS and MPLS traffic. The application cannot intercept Address Resolution Protocol (ARP) or other Layer 2 exception packets. The interception filter can be configured to timeout based on factors like total time (seconds), idle time (seconds), total packets or total data transmitted (bytes).

RELATED DOCUMENTATION

Configuring Dynamic Flow Capture | 284
Example: Configuring Dynamic Flow Capture on MX Series Routers | 292

Configuring Dynamic Flow Capture

IN THIS SECTION

Configuring the Capture Group

A capture group defines a profile for dynamic flow capture (DFC). The static configuration includes information about control sources, content destinations, and notification destinations. Dynamic configuration is added through interaction with control sources using a control protocol.

To configure a capture group, include the `capture-group` statement at the `[edit services dynamic-flow-capture]` hierarchy level:

```
capture-group client-name {
    content-destination identifier {
        address address;
        hard-limit bandwidth;
        hard-limit-target bandwidth;
        soft-limit bandwidth;
        soft-limit-clear bandwidth;
        ttl hops;
    }
    control-source identifier {
        allowed-destinations [ destinations ];
        minimum-priority value;
        no-syslog;
        notification-targets address port port-number;
        service-port port-number;
        shared-key value;
        source-addresses [ addresses ];
    }
    duplicates-dropped-periodicity seconds;
    input-packet-rate-threshold rate;
    interfaces interface-name;
    max-duplicates number;
    pic-memory-threshold percentage percentage;
}
```

To specify the `capture-group`, assign it a unique *`client-name`* that associates the information with the requesting control sources.

Configuring the Content Destination

You must specify a destination for the packets that match DFC PIC filter criteria. To configure the content destination, include the `content-destination` statement at the `[edit services dynamic-flow-capture capture-group client-name]` hierarchy level:

```
content-destination identifier {
    address address;
    hard-limit bandwidth;
    hard-limit-target bandwidth;
```

```
    soft-limit bandwidth;
    soft-limit-clear bandwidth;
    ttl hops;
}
```

Assign the `content-destination` a unique *`identifier`*. You must also specify its IP address and you can optionally include additional settings:

- `address`—The DFC PIC interface appends an IP header with this destination address on the matched packet (with its own IP header and contents intact) and sends it out to the content destination.

- `ttl`—The time-to-live (TTL) value for the IP-IP header. By default, the TTL value is 255. Its range is 0 through 255.

- Congestion thresholds—You can specify per-content destination bandwidth limits that control the amount of traffic produced by the DFC PIC during periods of congestion. The thresholds are arranged in two pairs: `hard-limit` and `hard-limit-target`, and `soft-limit` and `soft-limit-clear`. You can optionally include one or both of these paired settings. All four settings are 10–second average bandwidth values in bits per second. Typically `soft-limit-clear` < `soft-limit` < `hard-limit-target` < `hard-limit`. When the content bandwidth exceeds the `soft-limit` setting:

  1. A congestion notification message is sent to each control source of the criteria that point to this content destination

  2. If the control source is configured for `syslog`, a system log message is generated.

  3. A latch is set, indicating that the control sources have been notified. No additional notification messages are sent until the latch is cleared, when the bandwidth falls below the `soft-limit-clear` value.

  When the bandwidth exceeds the `hard-limit` value:

  1. The software begins deleting criteria until the bandwidth falls below the `hard-limit-target` value.

  2. For each criterion deleted, a CongestionDelete notification is sent to the control source for that criterion.

  3. If the control source is configured for `syslog`, a log message is generated.

  The application evaluates criteria for deletion using the following data:

  - Priority—Lower priority criteria are purged first, after adjusting for control source minimum priority.

  - Bandwidth—Higher bandwidth criteria are purged first.

  - Timestamp—The more recent criteria are purged first.

Configuring the Control Source

You configure information about the control source, including allowed source addresses and destinations and authentication key values. To configure the control source information, include the `control-source` statement at the `[edit services dynamic-flow-capture capture-group *client-name*]` hierarchy level:

```
control-source identifier {
    allowed-destinations [ destination-identifiers ];
    minimum-priority value;
    no-syslog;
    notification-targets address port port-number;
    service-port port-number;
    shared-key value;
    source-addresses [ addresses ];
}
```

Assign the `control-source` statement a unique *identifier*. You can also include values for the following statements:

- `allowed-destinations`—One or more content destination identifiers to which this control source can request that matched data be sent in its control protocol requests. If you do not specify any content destinations, all available destinations are allowed.

- `minimum-priority`—Value assigned to the control source that is added to the priority of the criteria in the DTCP ADD request to determine the total priority for the criteria. The lower the value, the higher the priority. By default, `minimum-priority` has a value of 0 and the allowed range is 0 through 254.

- `notification-targets`—One or more destinations to which the DFC PIC interface can log information about control protocol-related events and other events such as PIC bootup messages. You configure each `notification-target` entry with an IP `address` value and a User Datagram Protocol (UDP) `port` number.

- `service-port`—UDP port number to which the control protocol requests are directed. Control protocol requests that are not directed to this port are discarded by DFC PIC interfaces.

- `shared-key`—20-byte authentication key value shared between the control source and the DFC PIC monitoring platform.

- `source-addresses`—One or more allowed IP addresses from which the control source can send control protocol requests to the DFC PIC monitoring platform. These are /32 addresses.

Configuring the DFC PIC Interface

You specify the interface that interacts with the control sources configured in the same capture group. A Monitoring Services III PIC can belong to only one capture group, and you can configure only one PIC for each group.

To configure a DFC PIC interface, include the `interfaces` statement at the `[edit services dynamic-flow-capture capture-group client-name]` hierarchy level:

```
interfaces interface-name;
```

You specify DFC interfaces using the `dfc-` identifier at the `[edit interfaces]` hierarchy level. You must specify three logical units on each DFC PIC interface, numbered `0`, `1`, and `2`. You cannot configure any other logical interfaces.

- `unit 0` processes control protocol requests and responses.
- `unit 1` receives monitored data.
- `unit 2` transmits the matched packets to the destination address.

The following example shows the configuration necessary to set up a DFC PIC interface and intercept both IPv4 and IPv6 traffic:

```
[edit interfaces dfc-0/0/0]
unit 0 {
    family inet {
        filter {
            output high;  #Firewall filter to route control packets
            # through 'network-control' forwarding class. Control packets
            # are loss sensitive.
        }
        address 10.1.0.0/32 { # DFC PIC address
            destination 10.36.100.1; # DFC PIC address used by
            # the control source to correspond with the
            # monitoring platform
        }
    }
unit 1 { # receive data packets on this logical interface
    family inet; # receive IPv4 traffic for interception
    family inet6; # receive IPv6 traffic for interception
}
unit 2 { # send out copies of matched packets on this logical interface
```

```
    family inet;
}
```

In addition, you must configure dynamic flow capture to run on the DFC PIC in the correct chassis location. The following example shows this configuration at the `[edit chassis]` hierarchy level:

```
fpc 0 {
    pic 0 {
        monitoring-services application dynamic-flow-capture;
    }
}
```

Configuring the Firewall Filter

You can specify the firewall filter to route control packets through the network control forwarding class. The control packets are loss sensitive. To configure the firewall filter, include the following statements at the `[edit]` hierarchy level:

```
firewall {
    family inet {
        filter high {
            term all {
                then forwarding-class network-control;
            }
        }
    }
}
```

Configuring System Logging

By default, control protocol activity is logged as a separate system log facility, `dfc`. To modify the filename or level at which control protocol activity is recorded, include the following statements at the `[edit syslog]` hierarchy level:

```
file dfc.log {
    dfc any;
}
```

To cancel logging, include the `no-syslog` statement at the `[edit services dynamic-flow-capture capture-group` *`client-name`* `control-source` *`identifier`*`]` hierarchy level:

```
no-syslog;
```

NOTE: The dynamic flow capture (`dfc-`) interface supports up to 10,000 filter criteria. When more than 10,000 filters are added to the interface, the filters are accepted, but system log messages are generated indicating that the filter is full.

Configuring Tracing Options for Dynamic Flow Capture Events

You can enable tracing options for dynamic flow capture events by including the `traceoptions` statement at the `[edit services dynamic-flow-capture]` hierarchy level.

When you include the `traceoptions` configuration, you can also specify the trace file name, maximum number of trace files, the maximum size of trace files, and whether the trace file can be read by all users or not.

To enable tracing options for dynamic flow capture events, include the following configuration at the `[edit services dynamic-flow-capture]` hierarchy level:

```
traceoptions{
    file filename <files number> <size size> <world-readable | non-world-readable>;
}
```

To disable tracing for dynamic flow capture events, delete the `traceoptions` configuration from the `[edit services dynamic-flow-capture]` hierarchy level.

Configuring Thresholds

You can optionally specify threshold values for the following situations in which warning messages are recorded in the system log:

- Input packet rate to the DFC PIC interfaces
- Memory usage on the DFC PIC interfaces

To configure threshold values, include the `input-packet-rate-threshold` or `pic-memory-threshold` statements at the `[edit services dynamic-flow-capture capture-group client-name]` hierarchy level:

```
input-packet-rate-threshold rate;
pic-memory-threshold percentage percentage;
```

If these statements are not configured, no threshold messages are logged. The threshold settings are configured for the capture group as a whole.

The range of configurable values for the `input-packet-rate-threshold` statement is 0 through 1 Mpps. The PIC calibrates the value accordingly; the Monitoring Services III PIC caps the threshold value at 300 Kpps and the Multiservices 400 PIC uses the full configured value. The range of values for the `pic-memory-threshold` statement is 0 to 100 percent.

Limiting the Number of Duplicates of a Packet

You can optionally specify the maximum number of duplicate packets the DFC PIC is allowed to generate from a single input packet. This limitation is intended to reduce the load on the PIC when packets are sent to multiple destinations. When the maximum number is reached, the duplicates are sent to the destinations with the highest criteria class priority. Within classes of equal priority, criteria having earlier timestamps are selected first.

To configure this limitation, include the `max-duplicates` statement at the `[edit services dynamic-flow-capture capture-group client-name]` hierarchy level:

```
max-duplicates number;
```

You can also apply the limitation on a global basis for the DFC PIC by including the `g-max-duplicates` statement at the `[edit services dynamic-flow-capture]` hierarchy level:

```
g-max-duplicates number;
```

By default, the maximum number of duplicates is set to 3. The range of allowed values is 1 through 64. A setting for `max-duplicates` for an individual capture-group overrides the global setting.

In addition, you can specify the frequency with which the application sends notifications to the affected control sources that duplicates are being dropped because the threshold has been reached. You configure this setting at the same levels as the maximum duplicates settings, by including the `duplicates-dropped-periodicity` statement at the `[edit services dynamic-flow-capture capture-group client-name]` hierarchy level or the `g-duplicates-dropped-periodicity` statement at the `[edit services dynamic-flow-capture]` hierarchy level:

```
duplicates-dropped-periodicity seconds;
g-duplicates-dropped-periodicity seconds;
```

As with the `g-max-duplicates` statement, the `g-duplicates-dropped-periodicity` statement applies the setting globally for the application and is overridden by a setting applied at the capture-group level. By default, the frequency for sending notifications is 30 seconds.

RELATED DOCUMENTATION

Understanding Dynamic Flow Capture | 281
Example: Configuring Dynamic Flow Capture on MX Series Routers | 292

Example: Configuring Dynamic Flow Capture on MX Series Routers

The following example includes all parts of a complete dynamic flow capture configuration.

Configure the dynamic flow capture (DFC) PIC interface:

```
[edit interfaces dfc-0/0/0]
unit 0 {
    family inet {
        filter {
            output high;  #Firewall filter to route control packets
            # through 'network-control' forwarding class. Control packets
            # are loss sensitive.
        }
        address 10.1.0.0/32 { # DFC PIC address
            destination 10.36.100.1; # DFC PIC address used by
            # the control source to correspond with the
            # monitoring platform
        }
    }
unit 1 { # receive data packets on this logical interface
    family inet;
    family inet6;
```

```
}
unit 2 { # send out copies of matched packets on this logical interface
    family inet;
}
```

Configure the capture group:

```
services dynamic-flow-capture {
    capture-group g1 {
        interfaces dfc-0/0/0;
        input-packet-rate-threshold 90k;
        pic-memory-threshold percentage 80;
        control-source cs1 {
            source-addresses 10.36.41.1;
            service-port 2400;
            notification-targets {
                10.36.41.1 port 2100;
            }
            shared-key "$ABC123";
            allowed-destinations cd1;
        }
        content-destination cd1 {
            address 10.36.70.2;
            ttl 244;
        }
    }
}
```

Configure filter-based forwarding (FBF) to the dynamic flow capture PIC interface, logical unit 1.

For more information about configuring passive monitoring interfaces, see "Enabling Passive Flow Monitoring on M Series, MX Series or T Series Routers" on page 156.

```
interfaces so-1/2/0 {
    encapsulation ppp;
    unit 0 {
        passive-monitor-mode;
        family inet {
            filter {
                input catch;
            }
        }
```

```
    }
}
```

Configure the firewall filter:

```
firewall {
    filter catch {
        interface-specific;
        term def {
            then {
                count counter;
                routing-instance fbf_inst;
            }
        }
    }
    family inet {
        filter high {
            term all {
                then forwarding-class network-control;
            }
        }
    }
}
```

Configure a forwarding routing instance. The next hop points specifically to the logical interface corresponding to `unit 1`, because only this particular logical unit is expected to relay monitored data to the dynamic flow capture PIC.

```
routing-instances fbf_inst {
    instance-type forwarding;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop dfc-0/0/0.1;
        }
    }
}
```

Configure routing table groups:

```
[edit]
routing-options {
    interface-routes {
        rib-group inet common;
    }
    rib-groups {
        common {
            import-rib [ inet.0 fbf_inst.inet.0 ];
        }
    }
    forwarding-table {
        export pplb;
    }
}
```

Configure interfaces to the control source and content destination:

```
interfaces fe-4/1/2 {
    description "to cs1 from dfc";
    unit 0 {
        family inet {
            address 10.36.41.2/30;
        }
    }
}
interfaces ge-7/0/0 {
    description "to cd1 from dfc";
    unit 0 {
        family inet {
            address 10.36.70.1/30;
        }
    }
}
```

RELATED DOCUMENTATION

Understanding Dynamic Flow Capture | 281

Configuring Dynamic Flow Capture | 284

Monitoring a Capture Group Using SNMP or Show Services Commands

The Dynamic Flow Capture MIB provides a way to monitor dynamic flow capture information by using Simple Network Management Protocol (SNMP). The MIB provides the same information that you can view with the `show services dynamic-flow-capture content-destination`, `show services dynamic-flow-capture control-source`, and `show services dynamic-flow-capture statistics` commands.

CHAPTER 7

Detecting Threats and Intercepting Flows Using Junos Flow-Tap and FlowTapLite Services

IN THIS CHAPTER

Understanding the FlowTap and FlowTapLite Services

SUMMARY

Dynamic flow capture enables you to capture packet flows on the basis of dynamic filtering criteria, using Dynamic Tasking Control Protocol (DTCP) requests. The FlowTap and FlowTap Lite services are applications that perform lawful intercept of packet flows, using Dynamic Tasking Control Protocol (DTCP). The application extends the use of DTCP to intercept IPv4 and IPv6 packets in an active monitoring router and send a copy of packets that match filter criteria to one or more content destinations. FlowTap data provides flexible trend analysis for detection of new security threats.

FlowTap data can be used in the following applications:

- Flexible trend analysis for detection of new security threats
- Lawful intercept

NOTE: . For information about DTCP, see https://datatracker.ietf.org/doc/html/draft-cavuto-dtcp-03.

The FlowTap service is supported on MX Series routers. FlowTap filters are applied on all IPv4 traffic and do not add any perceptible delay in the forwarding path. FlowTap filters can also be applied on IPv6 traffic. For security, filters installed by one client are not visible to others and the CLI configuration does not reveal the identity of the monitored target. A lighter version of the application (FlowTapLite) is also available on ACX (Junos OS Evolved) and MX Series (Junos OS) routers. All of the FlowTapLite functionality resides in the Packet Forwarding Engine rather than in a service PIC or Dense Port Concentrator (DPC).

For detailed information about the FlowTap and FlowTapLite applications, see the following topics:

- "Understanding FlowTap and FlowTapLite Architecture" on page 298
- "Configuring a FlowTap Interface on MX Series Routers " on page 305
- "Configuring FlowTap and FlowTapLite Security Properties" on page 306
- "FlowTap and FlowTapLite Application Restrictions " on page 307

RELATED DOCUMENTATION

Configuring the FlowTap Service on MX Series Routers | **302**

Configuring FlowTapLite on MX Series Routers | **314**

Examples: Configuring the FlowTapLite Application on MX Series and ACX Series Routers | **309**

Understanding FlowTap and FlowTapLite Architecture

SUMMARY

Learn about mediation devices and how they interact with the router running the FlowTap or FlowTapLite application.

The flow-tap architecture consists of one or more *mediation devices* that send requests to an ACX or MX Series router that supports this feature to monitor incoming data. Any packets that match specific filter criteria are forwarded to a set of one or more *content destinations*:

- Mediation device—A client that monitors electronic data or voice transfer over the network. The mediation device sends filter requests to the router using DTCP. The clients are not identified for security reasons, but have permissions defined by a set of special login classes.

- Monitoring platform—A router configured to support the FlowTap or FlowTapLite application. The monitoring platform processes the requests from the mediation devices, applies the dynamic filters, monitors incoming data flows, and sends the matched packets to the appropriate content destinations.

- Content destination—Recipient of the matched packets from the monitoring platform. Typically the matched packets are sent using an IP Security (IPSec) tunnel from the monitoring platform to another router connected to the content destination. The content destination and the mediation device can be physically located on the same host.

- Dynamic filters—The Packet Forwarding Engine automatically generates a *firewall filter* that is applied to all IPv4 routing instances. Each term in the filter includes a `flow-tap` action that is similar to the existing `sample` or `port-mirroring` actions. As long as one of the filter terms matches an incoming packet, the router copies the packet and forwards it to the MPC card or line card that is configured for `flow-tap` service. The card runs the packet through the client filters and sends a copy to each matching content destination. For security, filters installed by one client are not visible to others and the CLI configuration does not reveal the identity of the monitored target.

  Following is a sample filter configuration; note that it is dynamically generated by the router (no user configuration is required):

```
filter combined_LEA_filter {
    term LEA1_filter {
        from {
            source-address 192.0.2.1;
            destination-address 198.51.100.6;
        }
        then {
            flow-tap;
        }
    }
    term LEA2_filter {
        from {
            source-address 10.1.1.1;
            source-port    23;
        }
```

```
            then {
                flow-tap;
            }
        }
    }
```

Figure 26 on page 300 shows a sample topology that uses two mediation devices and two content destinations.

Figure 26: MX Series FlowTap Topology Diagram

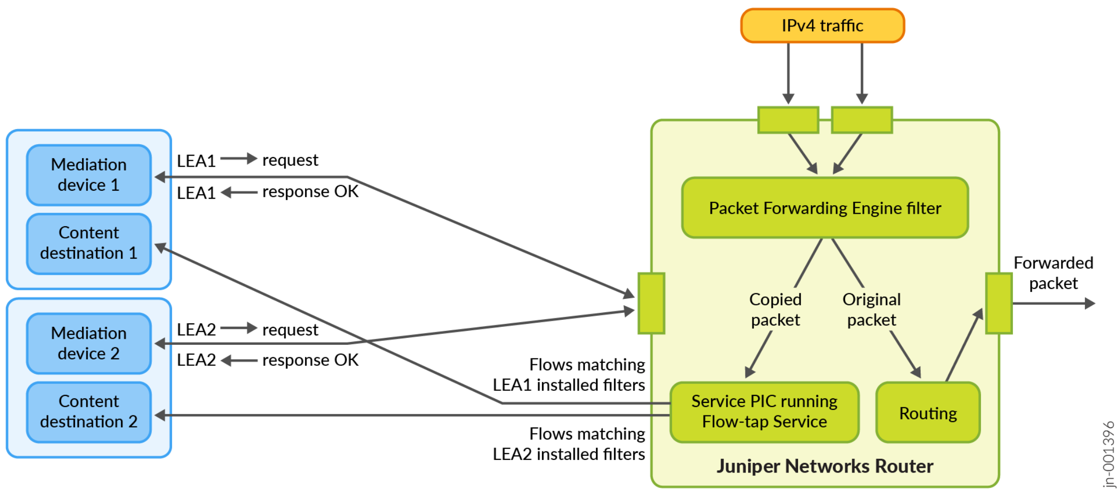
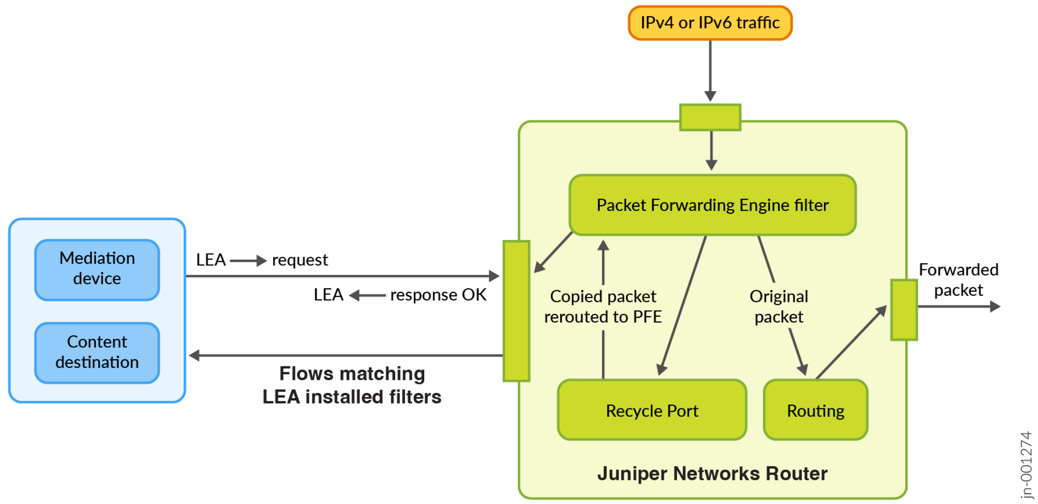


Figure 27 on page 301 shows a sample topology that uses one mediation device and one content destination. Note that the ACX Series routers use a recycle port.

Figure 27: ACX Series FlowTapLite Topology Diagram



RELATED DOCUMENTATION

Configuring the FlowTap Service on MX Series Routers

IN THIS SECTION

This topic explains FlowTap configuration.

Configuring the FlowTap Interface

To configure an adaptive services interface for FlowTap service, include the `interface` statement at the `[edit services flow-tap]` hierarchy level:

```
interface sp-fpc/pic/port.unit-number;
```

You can assign any Adaptive Services or Multiservices PIC in the active monitoring router and use any logical unit on the PIC.

You can specify the type of traffic for which you want to apply the FlowTap service by including the `family inet | inet6` statement. If the `family` statement is not included, the FlowTap service is, by default, applied to the IPv4 traffic. To apply FlowTap service to IPv6 traffic, you must include the `family inet6` statement in the configuration. To enable the FlowTap service for IPv4 and IPv6 traffic, you must explicitly configure the `family` statement for both the `inet` family and the `inet6` family.

NOTE: You cannot configure dynamic flow capture and FlowTap services on the same router simultaneously.

You must also configure the logical interface at the `[edit interfaces]` hierarchy level:

```
interface sp-fpc/pic/port {
    unit logical-unit-number {
        family inet;
        family inet6;
    }
}
```

NOTE: If you do not include the `family inet6` statement in the configuration, IPv6 flows are not intercepted. Note that the FlowTap solution did not support IPv6.

Firewall Filters and Tapping

Tapping happens no matter what firewall filter action is configured for the packets arriving on the ingress interface; that is, tapping happens even when the firewall filter action is set to discard or reject the packets, with one exception—tapping does not happen when the ingress interface has the firewall filter action as reject and the DTCP filter is for tapping **all v4** traffic. Tapping does happen when the ingress interface has the firewall filter action as reject and the DTCP filter is for tapping **input interface** traffic.

Strengthening FlowTap Security

You can add an extra level of security to Dynamic Tasking Control Protocol (DTCP) transactions between the mediation device and the router by enabling DTCP sessions on top of the SSH layer. To configure SSH settings, include the `flow-tap-dtcp` statement at the `[edit system services]` hierarchy level:

```
flow-tap-dtcp {
    ssh {
        connection-limit value;
        rate-limit value;
    }
}
```

To configure client permissions for viewing and modifying FlowTap configurations and for receiving tapped traffic, include the `permissions` statement at the `[edit system login class class-name]` hierarchy level:

```
permissions [permissions];
```

The permissions needed to use FlowTap features are as follows:

- `flow-tap`—Can view FlowTap configuration
- `flow-tap-control`—Can modify FlowTap configuration
- `flow-tap-operation`—Can tap flows

You can also specify user permissions on a RADIUS server, for example:

```
Bob Auth-Type := Local, User-Password = = “abc123”
Juniper-User-Permissions = “flow-tap-operation”
```

MX Series routers that support this feature can process mediation device DTCP ADD requests that contain up to 15 source-destination port pairs. Multiple source-destination port pairs must be separated by commas. For example:

```
ADD DTCP/0.7
Csource-ID: ftap
Cdest-ID: cd2
Source-Port: 2000,8001,4000,5000,6000,6001,6002
Dest-Port: 2000,9001,4000,5000,6000,9000
```

For details on `[edit system]` and RADIUS configuration, see the User Access and Authentication Administration Guide for Junos OS.

Restrictions on FlowTap Services

The following restrictions apply to Junos FlowTap services:

- You cannot configure dynamic flow capture and FlowTap services on the same router simultaneously.
- On routers that support LMNR-based FPCs, you cannot configure the FlowTap service for IPv6 along with port mirroring or sampling of IPv6 traffic. This restriction applies even if the router does not have any LMNR-based FPC installed in it. However, there is no restriction on configuring the FlowTap service on routers that are configured for port mirroring or sampling of IPv4 traffic.
- FlowTap does not support interception of MPLS and virtual private LAN service (VPLS).

- FlowTap cannot intercept Address Resolution Protocol (ARP) and other Layer 2 exceptions.
- IPv4 and IPv6 intercept filters can coexist on a system, subject to a combined maximum of 100 filters.
- When the dynamic flow capture process or the Adaptive Services or Multiservices PIC configured for FlowTap restarts, all filters are deleted and the mediation devices are disconnected.
- Only the first fragment of an IPv4 fragmented packet stream is sent to the content destination.
- Port mirroring might not work in conjunction with the FlowTap service.
- Running the FlowTap service over an IPsec tunnel on the same router can cause packet loops and is not supported.
- You cannot configure the FlowTap service on channelized interfaces.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
16.2	MX Series routers can process mediation device DTCP ADD requests that contain up to 15 source-destination port pairs. Multiple source-destination port pairs must be separated by commas.

RELATED DOCUMENTATION

Configuring FlowTapLite on MX Series Routers | 314

Configuring a FlowTap Interface on MX Series Routers

To configure an AS PIC interface for the FlowTap service, include the `interface` statement at the [**edit services flow-tap**] hierarchy level:

```
interface sp-fpc/pic/port.unit-number;
```

You can assign any AS PIC in the active monitoring station for FlowTap service, and use any logical unit on the PIC.

NOTE: You cannot configure dynamic flow capture and FlowTap features on the same router simultaneously.

You must also configure the logical interface at the `[edit interfaces]` hierarchy level:

```
interface sp-fpc/pic/port {
    unit logical-unit-number {
        family inet;
    }
}
```

RELATED DOCUMENTATION

Configuring FlowTap and FlowTapLite Security Properties

SUMMARY

Enable DTCP on top of the SSH layer and set permissions needed to configure and view FlowTap and FlowTapLite features.

You can add an extra level of security to DTCP transactions between the mediation device and the router by enabling DTCP sessions on top of the SSH layer. To configure, include the `flow-tap-dtcp` statement at the `[edit system services]` hierarchy level:

```
flow-tap-dtcp {
    ssh {
        connection-limit value;
```

```
        rate-limit value;
    }
}
```

To configure client permissions for viewing and modifying flow-tap configurations and for receiving tapped traffic, include the `permissions` statement at the `[edit system login class class-name]` hierarchy level:

```
permissions [ permissions ];
```

The permissions needed to use FlowTap and FlowTapLite features are as follows:

- `flow-tap`—Can view FlowTap and FlowTapLite configuration.
- `flow-tap-control`—Can modify FlowTap and FlowTapLite configuration.
- `flow-tap-operation`—Can tap flows.

You can also specify user permissions on a RADIUS server, for example:

```
Bob Auth-Type := Local, User-Password = = “abc123”
Juniper-User-Permissions = “flow-tap-operation”
```

RELATED DOCUMENTATION

Understanding FlowTap and FlowTapLite Architecture | 298
Configuring a FlowTap Interface on MX Series Routers | 305
FlowTap and FlowTapLite Application Restrictions | 307

FlowTap and FlowTapLite Application Restrictions

SUMMARY

The application restrictions vary by application and by router platform.

The following restrictions apply to FlowTap and FlowTapLite services:

- You cannot configure dynamic flow capture and FlowTap services on the same router simultaneously.
- When the dynamic flow capture process or an AS PIC configured for FlowTap processing restarts, all filters are deleted and the mediation devices are disconnected.
- Only the first fragment of an IPv4 fragmented packet stream is sent to the content destination.
- If the FlowTap application is configured, you cannot configure the filter action **then syslog** for any *firewall filter* running on the same platform.
- Running the FlowTap or FlowTapLite application over an IPsec tunnel on the same router can cause packet loops and is not supported.
- The FlowTapLite service `[edit services flow-tap]` on tunnel interfaces on supported routers and the RADIUS flow-tap service `[edit services radius-flow-tap]` cannot run simultaneously on the router. Consequently, you cannot run both FlowTapLite and subscriber secure policy mirroring at the same time on the same router in the earlier releases. FlowTapLite and subscriber secure policy mirroring are supported to run concurrently on the same MX Series router.
- For those ACX routers that support FlowTapLite:
  - You can configure only one mediation device in the system. That is, we support only one combination of source-ip/destination-ip/source-port/destination-port per mediation device tunnel. In contrast, MX Series devices can support more than one mediation device.
  - Ingress flow analyzers, ingress flow-mirroring, and egress flow-mirroring do not work for the flow-tapped packets.
  - You can match on only these three items: source IP address, destination IP address, and VRF.
  - If you want to tap by both source IP address and destination IP address, you must have two flow-tap rules, one for the source address and one for the destination address.
  - The configured IPv4 address can only be 32 bytes and the configured IPv6 address can only be 128 bytes; that is, network prefixes are not allowed.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.3R1	FlowTapLite and subscriber secure policy mirroring are supported to run concurrently on the same MX Series router.

RELATED DOCUMENTATION

Examples: Configuring the FlowTapLite Application on MX Series and ACX Series Routers

IN THIS SECTION

Configuration examples

This section contains two examples for the FlowTapLite application, one for the MX Series routers and one for the ACX routers that run Junos OS Evolved. The main differences between the two is the configuration for the tunnel interface, and that the ACX router requires you to configure a recycle port whereas the MX router does not require a recycle port.

See the figures below for topologies for the MX Series and the ACX Series FlowTapLite application:

Figure 28: MX Series FlowTapLite Topology

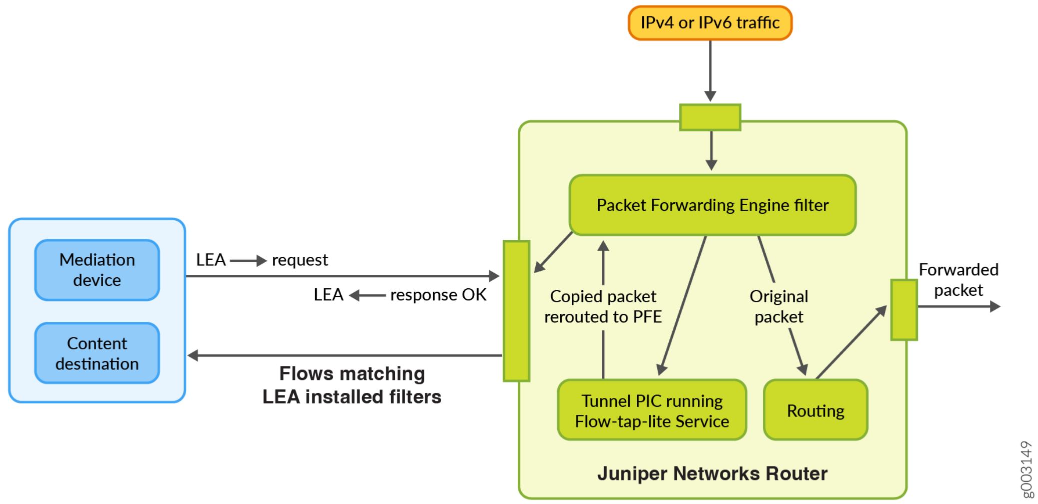
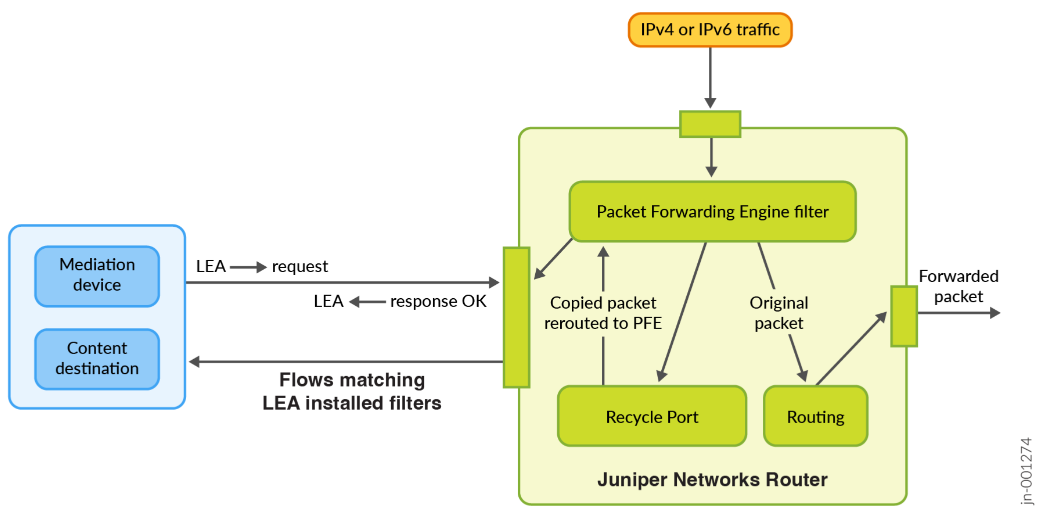


Figure 29: ACX Series FlowTapLite Topology



For the MX Series routers that support the FlowTapLite application, the following example shows a FlowTapLite configuration that intercepts IPv4 and IPv6 flows:

```
system {
    login {
        class flowtap {
```

```
            permissions flow-tap-operation;
        }
        user ftap {
            uid 2000;
            class flowtap;
            authentication {
                encrypted-password "$ABC123"; ## SECRET-DATA
            }
        }
    }
    services {
        flow-tap-dtcp {
            ssh;
        }
    }
}
    chassis {
        fpc 0 {
            pic 0 {
                tunnel-services {
                    bandwidth 10g;
                }
            }
        }
    }
    interfaces {
        vt-0/0/0 {
            unit 0 {
                family inet;
                family inet6;
            }
        }
    }
    services {
        flow-tap {
            tunnel-interface vt-0/0/0.0;
        }
    }
```

For the ACX routers running Junos OS Evolved that support the FlowTapLite application, the following example shows a FlowTapLite configuration that intercepts IPv4 and IPv6 flows:

```
[edit]
system {
    login {
        class flowtap {
            permissions flow-tap-operation;
        }
        user ftap {
            uid 2000;
            class flowtap;
            authentication {
                encrypted-password "$ABC123"; ## SECRET-DATA
            }
        }
    }
    packet-forwarding-options {
        recycle-bandwidth-profiles {
            p1 {
                fti-virtual-loop 1..100;
            }
        }
    }
    services {
        flow-tap-dtcp {
            ssh;
        }
    }
}
interfaces {
    fti2 {
        unit 0 {
            tunnel {
                encapsulation virtual-loop;
            }
            family inet;
            family inet6;
        }
    }
}
services {
```

```
    flow-tap {
        tunnel-interface fti2.0;
    }
}
```

DTCP Request examples

This section contains examples of DTCP Requests for ACX Series routers.

A DTCP Add Request:

```
ADD DTCP/0.8
Csource-ID: user-name
Cdest-ID: variable
Source-Address: ipv4-address
Dest-Address: ipv4-address
Source-Port: *
Dest-Port: *
Protocol: *
Priority: priority-number
X-JTap-Cdest-Dest-Address: ipv4-address
X-JTap-Cdest-Dest-Port: udp-port
X-JTap-Cdest-Source-Address: ipv4-address
X-JTap-Cdest-Source-Port: udp-port
X-JTap-Cdest-TTL: time-to-live
Seq: sequence-number
Flags: flag
```

A DTCP List Request:

```
LIST DTCP/0.8
Csource-ID: user-name
Cdest-ID: variable
Flags: BOTH
Seq: sequence-number
```

A DTCP Delete Request:

```
DELETE DTCP/0.8
Csource-ID: user-name
```

```
Cdest-ID: variable
Flags: flag
Seq: sequence-number
```

RELATED DOCUMENTATION

Configuring FlowTapLite on MX Series Routers

FlowTapLite, which is a lighter version of the FlowTap application, is available on MX Series routers. All of the functionality resides in the Packet Forwarding Engine rather than in a service PIC or Dense Port Concentrator (DPC). To see which routers and line cards support this feature, see Support for FlowTapLite.

FlowTapLite supports the sampling of circuit cross connect (CCC) traffic. DTCP/0.8 is required to specify `X-JTap-Filter-Family ccc`. L3 parameters cannot be included in a DTCP/0.8 ADD request containing `X-JTap-Filter-Family`. To see which routers support this feature, seeFlowTapLite support for circuit cross connect traffic.

FlowTapLite uses the same DTCP-SSH architecture to install the Dynamic Tasking Control Protocol (DTCP) filters and authenticate the users as the original flow-tap application and supports up to 3000 filters per chassis.

NOTE: The original FlowTap application and FlowTapLite cannot be used at the same time.

To configure FlowTapLite, include the `flow-tap` statement at the `[edit services]` hierarchy level:

```
flow-tap {
    tunnel-interface interface-name;
}
```

If you do not specify a family, FlowTapLite is applied only to IPv4 traffic. FlowTapLite can be applied to circuit cross connect traffic (ccc). DTCP/0.8 is required to specify X-JTap-Filter-Family ccc. L3 parameters cannot be included in a DTCP/0.8 ADD request containing X-JTap-Filter-Family.

For the Packet Forwarding Engine to encapsulate the intercepted packet, it must send the packet to a tunnel logical (`vt-`) interface. You need to allocate a tunnel interface and assign it to the dynamic flow capture process for FlowTapLite to use. To create the tunnel interface, include the following configuration:

```
chassis {
    fpc number {
        pic number {
            tunnel-services {
                bandwidth (1g | 10g);
            }
        }
    }
}
```

NOTE: Currently FlowTapLite supports only one tunnel interface per instance.

To configure the logical interfaces and assign them to the dynamic flow capture process, include the following configuration:

```
interfaces {
    vt-fpc/pic/port {
        unit 0 {
            family inet;
            family inet6;
        }
    }
}
```

NOTE: If a service PIC is available, you can use its tunnel interface for the same purpose.

NOTE: If you do not include the `family inet6` statement in the configuration, IPv6 flows are not intercepted.

NOTE: With FlowTapLite configured and traceoptions enabled, if you add more than two content destinations by including the X-JTAP- CDEST-DEST-ADDRESS line in the Dynamic Tasking Control Protocol (DTCP) parameter file and initiate a DTCP session by sending a DTCP ADD message, a `400 BAD request` message is received. Although you can specify more than two content destinations in the DTCP file that is sent from the mediation device, this error message occurs when the DTCP ADD message is sent. This behavior is expected with more than two content destinations. You must specify only two content destinations per DTCP ADD message.

FlowTapLite and subscriber secure policy mirroring are supported to run concurrently on the same MX Series router. To see which routers support this feature, see: Concurrent subscriber secure policy and FlowTapLite.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.3R1	FlowTapLite and subscriber secure policy mirroring are supported to run concurrently on the same MX Series router.
17.2R1	FlowTapLite supports the sampling of circuit cross connect (CCC) traffic.

RELATED DOCUMENTATION

Understanding the FlowTap and FlowTapLite Services | 297

Configuring the FlowTap Service on MX Series Routers | 302

Examples: Configuring the FlowTapLite Application on MX Series and ACX Series Routers | 309

Subscriber Secure Policy Overview

3

PART

Inline Monitoring Services and Inband Network Telemetry

CHAPTER 8

Inline Monitoring Services

IN THIS CHAPTER

Inline Monitoring Services Configuration

IN THIS SECTION

Understanding Inline Monitoring Services

IN THIS SECTION

Benefits of Inline Monitoring Services

Flexible—Inline monitoring services allow different inline-monitoring instances to be mapped to different firewall filter terms, unlike in traditional sampling technologies, where all the instances are mapped to the Flexible PIC Concentrator (FPC). This provides you with the flexibility of sampling different streams of traffic at different rates on a single interface.

Packet format agnostic—Traditional flow collection technologies rely on packet parsing and aggregation by the network element. With inline monitoring services, the packet header is exported to the collector for further processing, but without aggregation. Thereby, you have the benefit of using arbitrary packet fields to process the monitored packets at the collector.

Inline Monitoring Services Feature Overview

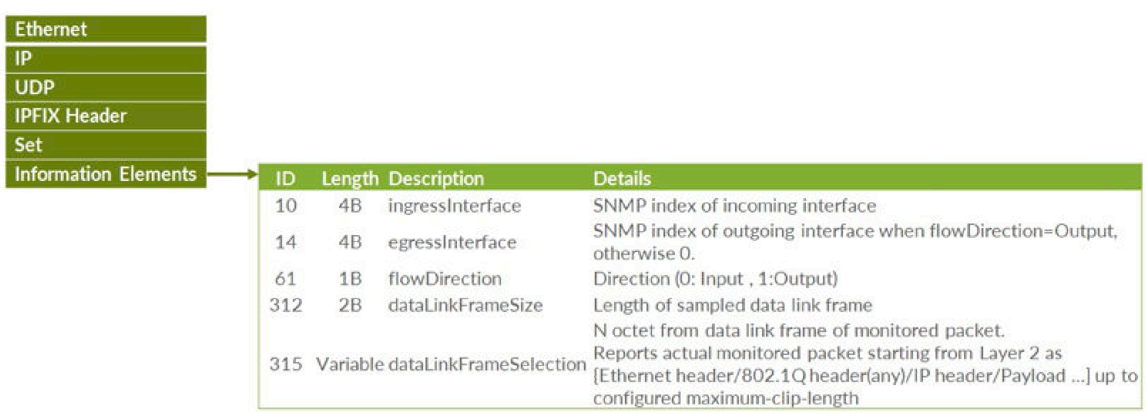
Service providers and content providers typically require visibility into traffic flows to evaluate peering agreements, detect traffic anomalies and policy violations, and monitor network performance. To meet these requirements, you would traditionally export aggregate flow statistics information using JFlow or IPFIX variants.

As an alternative approach, you can sample the packet content, add metadata information, and export the monitored packets to an collector. The inline monitoring services enable you to do this on MX Series routers and on PTX routers that run Junos OS Evolved.

With inline monitoring services, you can monitor every IPv4 and IPv6 packet on both ingress and egress directions of an interface. The software encapsulates the monitored traffic in an IPFIX format and exports the actual packet up to the configured clip length to an collector for further processing. By default, Junos OS supports a maximum clip length of 126 bytes starting from the Ethernet header and Junos OS Evolved supports a maximum clip length of 256 bytes starting from the Ethernet header.

Figure 30 on page 319 illustrates the IPFIX format specification.

Figure 30: Inline Monitoring IPFIX Specification



ID	Length	Description	Details
10	4B	ingressInterface	SNMP index of incoming interface
14	4B	egressInterface	SNMP index of outgoing interface when flowDirection=Output, otherwise 0.
61	1B	flowDirection	Direction (0: Input , 1:Output)
312	2B	dataLinkFrameSize	Length of sampled data link frame
315	Variable	dataLinkFrameSelection	N octet from data link frame of monitored packet. Reports actual monitored packet starting from Layer 2 as [Ethernet header/802.1Q header(any)/IP header/Payload ...] up to configured maximum-clip-length

The IPFIX header and IPFIX payload are encapsulated using IP or UDP transport layer. The exported IPFIX format includes two data records and two data templates that are exported to every collector:

- Data record—Includes incoming and outgoing interface, flow direction, data link frame section, and data link frame size. This information is sent to the collector only when sampled packets are being exported.

  Figure 31 on page 321 is a sample illustration of IPFIX data record packet.

- Option data record—Includes system level information, such as exporting process ID, and sampling interval. This information is sent to the collector periodically, irrespective of whether sampling packets are being exported are not.

  Figure 32 on page 321 is a sample illustration of IPFIX option data record packet.

Table 48: Information Element fields in IPFIX Option Data Packet

Number	Information Element ID	Information Element Length	Details
1	144	4B	Observation domain ID - An unique identifier of exporting process per IPFIX device. Purpose of this field is to limit the scope of other information element fields.
2	34	4B	Sampling interval at which the packets are sampled. 1000 indicates that one of 1000 packets is sampled.

- Data template—Includes five information elements:
  - Ingress interface
  - Egress interface
  - Flow direction
  - Data link frame size
  - Variable data link frame selection

  Figure 33 on page 322 is a sample illustration of IPFIX data template packet.

- Option data template—Includes flow exporter and sampling interval information.

  Figure 34 on page 322 is a sample illustration of IPFIX option data template packet.

When there is a new or changed inline monitoring services configuration, periodic export of data template and option data template is immediately sent to the respective collectors.

Figure 31: IPFIX Data Record

```
  Version: 10
  Length: 160
▶ Timestamp: Feb 28, 2019 14:05:41.000000000 IST
  FlowSequence: 474
  Observation Domain Id: 1342242816
▼ Set 1 [id=2000] (1 flows)
    FlowSet Id: (Data) (2000)
    FlowSet Length: 144
    [Template Frame: 9]
  ▼ Flow 1
      InputInt: 553
      OutputInt: 0
      Direction: Ingress (0)
      Data Link Frame Size: 1496
    ▼ Data Link Frame Section: 00711f7ce252000001000e000800450005ca000000004011...
        String_len_short: 128
```

Figure 32: IPFIX Option Data Record

```
  Version: 10
  Length: 28
▶ Timestamp: Feb 28, 2019 14:21:10.000000000 IST
  FlowSequence: 11
  Observation Domain Id: 1342242816
▼ Set 1 [id=2600] (1 flows)
    FlowSet Id: (Data) (2600)
    FlowSet Length: 12
    [Template Frame: 1]
  ▼ Flow 1
      FlowExporter: 1
      Sampling interval: 1
```

Figure 33: IPFIX Data Template

```
  Version: 10
  Length: 44
▸ Timestamp: Feb 28, 2019 14:05:42.000000000 IST
  FlowSequence: 474
  Observation Domain Id: 1342242816
▾ Set 1 [id=2] (Data Template): 2000
    FlowSet Id: Data Template (V10 [IPFIX]) (2)
    FlowSet Length: 28
  ▾ Template (Id = 2000, Count = 5)
      Template Id: 2000
      Field Count: 5
    ▾ Field (1/5): INPUT_SNMP
        0... .... .... .... = Pen provided: No
        .000 0000 0000 1010 = Type: INPUT_SNMP (10)
        Length: 4
    ▾ Field (2/5): OUTPUT_SNMP
        0... .... .... .... = Pen provided: No
        .000 0000 0000 1110 = Type: OUTPUT_SNMP (14)
        Length: 4
    ▾ Field (3/5): DIRECTION
        0... .... .... .... = Pen provided: No
        .000 0000 0011 1101 = Type: DIRECTION (61)
        Length: 1
    ▾ Field (4/5): dataLinkFrameSize
        0... .... .... .... = Pen provided: No
        .000 0001 0011 1000 = Type: dataLinkFrameSize (312)
        Length: 2
    ▾ Field (5/5): dataLinkFrameSection
        0... .... .... .... = Pen provided: No
        .000 0001 0011 1011 = Type: dataLinkFrameSection (315)
        Length: 65535 [i.e.: "Variable Length"]
```

Figure 34: IPFIX Option Data Template

```
  Version: 10
  Length: 36
▸ Timestamp: Feb 28, 2019 14:21:10.000000000 IST
  FlowSequence: 11
  Observation Domain Id: 1342242816
▾ Set 1 [id=3] (Options Template): 2600
    FlowSet Id: Options Template (V10 [IPFIX]) (3)
    FlowSet Length: 20
  ▾ Options Template (Id = 2600) (Scope Count = 1; Data Count = 1)
      Template Id: 2600
      Total Field Count: 2
      Scope Field Count: 1
    ▾ Field (1/1) [Scope]: FLOW_EXPORTER
        0... .... .... .... = Pen provided: No
        .000 0000 1001 0000 = Type: FLOW_EXPORTER (144)
        Length: 4
    ▾ Field (1/1): SAMPLING_INTERVAL
        0... .... .... .... = Pen provided: No
        .000 0000 0010 0010 = Type: SAMPLING_INTERVAL (34)
        Length: 4
    Padding: 0000
```

Inline Monitoring Services Configuration Overview

You can configure a maximum of sixteen (Junos OS) or seven (Junos OS Evolved) inline-monitoring instances that support template and collector-specific configuration parameters. Each inline monitoring instance supports up to four collectors (maximum of 64 collectors in total), and, for Junos OS only, you can specify different sampling rates under each collector configuration. Because of this flexibility, the inline monitoring services overcome the limitations of traditional sampling technologies, such as JFlow, sFlow, and port mirroring.

To configure inline monitoring:

1. You must include the `inline-monitoring` statement at the `[edit services]` hierarchy level. Here you specify the template and inline monitoring instance parameters. You must specify the collector parameters under the inline-monitoring instance.

2. Specify arbitrary match conditions using a firewall filter term and an action to accept the configured inline-monitoring instance. This maps the inline-monitoring instance to the firewall term.

3. Map the firewall filter under the family `inet` or `inet6` statement using the `inline-monitoring-instance` statement at the [edit firewall filter *name* then] hierarchy level. Starting in Junos OS Release 21.1R1, you can also map the firewall filter under the family `any`, `bridge`, `ccc`, `mpls`, or `vpls` statements. For Junos OS Evolved, the `bridge` and `vpls`families are not supported; use the `ethernet-switch` family instead. Junos OS Evolved does support the `any`, `ccc`, `inet`, `inet6`, and `mpls` families as well. You can also alternatively apply the firewall filter to a forwarding table filter with input or output statement to filter ingress or egress packets, respectively.

Remember:

- The device must support a maximum packet length (clip length) of 126 bytes (Junos OS) or 256 bytes (Junos OS Evolved) to enable inline monitoring services.

- You cannot configure more than 16 (Junos OS) or 7 (Junos OS Evolved) inline-monitoring instances because of the scarcity of bits available in the packet in the forwarding path.

- Apply inline monitoring services only on a collector interface, that is, the interface on which the collector is reachable. You must not apply inline monitoring on IPFIX traffic as this generates another IPFIX packet for sampling, thereby creating a loop. This includes inline monitoring service-generated traffic, such as template and record packets, option templates, and option record packets.

- When inline monitoring service is enabled on aggregated Ethernet (AE) interfaces, the information element values are as follows:

Table 49: Information Element Values for Aggregated Ethernet Interfaces

Direction of inline monitoring service on AE interface	Information element-10 (Incoming interface)	Information element-14 (Outgoing interface)
Ingress	SNMP ID of AE	0
Egress	SNMP ID of AE	SNMP ID of member link

- When inline monitoring service is enabled on IRB interfaces, the information element values are as follows:

  Table 50: Information Element Values for IRB Interfaces

Direction of inline monitoring service on IRB interface	Information element-10 (Incoming interface)	Information element-14 (Outgoing interface)
Ingress	SNMP ID of IRB	0
Egress	SNMP ID of IRB	SNMP ID of vlan-bridge encapsulated interface

- For XL-XM based devices (with Lookup chip (XL) and buffering ASIC (XM)), the length of the Data Link Frame Section information element in an exported packet can be shorter than the clip length even if the egress packet length is greater than clip length.

  The length of the Data Link Frame Section information element is reduced by 'N' number of bytes where 'N' = (ingress packet Layer 2 encapsulation length - egress packet Layer 2 encapsulation length).

  For instance, the Layer 2 encapsulation length for the ingress packet is greater than that of the egress packet when the ingress packet has MPLS labels and egress packet is of IPv4 or IPv6 type. When traffic flows from the provider edge (PE) device to the customer edge (CE) device, the ingress packet has VLAN tags and the egress packet is untagged.

  In such cases, the clip length can go past the last address location of the packet head, generating a `PKT_HEAD_SIZE` system log message. This can result in degradation of packet forwarding for the device.

- In case of inline monitoring services in the ingress direction, the `egressInterface` (information element ID 14) does not report SNMP index of the output interface. This information element ID always reports value zero in case of ingress direction. The receiving collector process should identify the validity of this field based on the `flowDirection` (information element ID 61).

Supported and Unsupported Features with Inline Monitoring Services

Inline monitoring services supports:

- Graceful Routing Engine switchover
- In-service software upgrade (ISSU), nonstop software upgrade (NSSU), and nonstop active routing (NSR)
- Ethernet interfaces and integrated routing and bridging (IRB) interfaces
- Junos node slicing
- Starting in Junos OS Evolved Release 22.4R1, configuring DSCP, forwarding class, or routing instances for collectors.
- Starting in Junos OS Evolved Release 22.4R1, configuring template IDs or option template IDs.

Inline monitoring services currently does not support:

- Configuring more than 16 (Junos OS) or 7 (Junos OS Evolved) inline-monitoring instances.
- Junos Traffic Vision
- Prior to Junos OS Release 21.1R1, the inline-monitoring-instance term action is supported only for `inet` and `inet6` family firewall filters. Starting in Junos OS Release 21.1R1, it is supported for the `any`, `bridge`, `ccc`, `mpls`, and `vpls` family firewall filters.
- IPv6 addressable collectors
- Virtual platforms
- Logical systems
- Configuring both the observation domain ID and observation cloud ID. You must choose only one of them.
- An inline monitoring instance action used for exception reporting cannot be used for any other purpose, such as a firewall re-direct action or a regular inline-monitoring action.
- An inline monitoring instance used for a firewall re-direct action cannot be used for any other purpose, such as exception reporting or a regular inline-monitoring action.
- Prior to Junos OS Evolved Release 22.4R1, configuring DSCP, forwarding class, or routing instances for collectors.

- Prior to Junos OS Evolved Release 22.4R1, configuring template IDs or option template IDs. The system generates these for you.
- Configuring port mirroring and inline monitoring services under the same firewall filter term (Junos OS Evolved).
- In the egress direction, configuring both SFlow and exception reporting; you must choose only one of them (Junos OS Evolved).

Configuring Inline Monitoring Services

The inline monitoring services can monitor both IPv4 and IPv6 traffic on both ingress and egress directions. You can enable inline monitoring on MX Series routers with MPCs (Junos OS) and on PTX routers that run Junos OS Evolved.

You can configure inline monitoring services to monitor different streams of traffic at different sampling rates on the same logical unit of the interface. You can also export the original packet size to an collector along with information on the interface origin for effective troubleshooting.

Before You Configure

When you configure inline monitoring services, you can:

- Configure up to 16 (Junos OS) or 7 (Junos OS Evolved) inline-monitoring instances. Under each instance, you can configure specific collector and template parameters.
- Configure up to 4 IPv4-addressable collectors under each inline-monitoring instance. In total, you can configure up to 64 collectors. The collectors can be remote, and at different locations.

  For each collector, you can configure specific parameters, such as source and destination address, and so on. The default routing-instance name at the collector is `default.inet`.
- For Junos OS, you can configure the `inet` or `inet6` family firewall filter with the term action `inline-monitoring-instance` *`inline-monitoring-instance-name`* . Starting in Junos OS Release 21.1R1, you can configure `any`, `bridge`, `ccc`, `mpls`,or `vpls` family firewall filters with the term action `inline-monitoring-instance` *`inline-monitoring-instance-name`* . For Junos OS Evolved, you can configure the `any`, `ccc`, `ethernet-switch`, `inet`, `inet6`, or `mpls` family firewall filters with the term action inline-monitoring-instance *inline-monitoring-instance-name*.

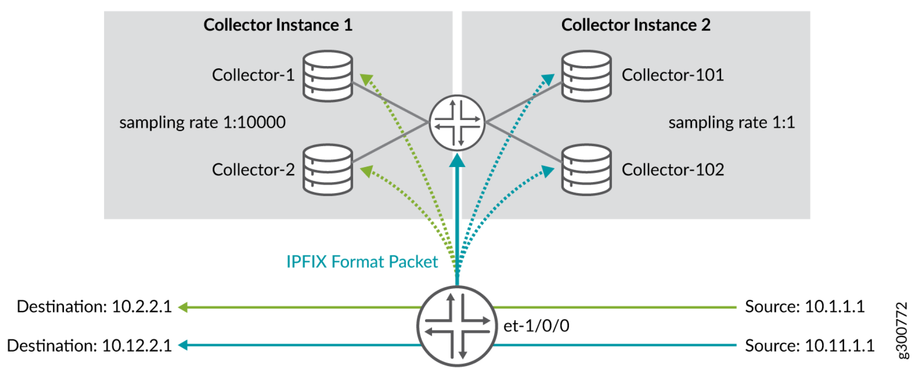
  Each term can support a different inline-monitoring instance.
- Attach the inline monitoring firewall filter under the family of the logical unit of the interface.

After successfully committing the configuration, you can verify the implementation of the inline monitoring services by issuing the *show services inline-monitoring statistics fpc-slot* command from the CLI.

NOTE: If a packet requires inline monitoring services to be applied along with any of the traditional sampling technologies (such as JFlow or SFlow), the Packet Forwarding Engine performs both inline monitoring services and the traditional sampling technology on that packet. Port mirroring currently must be configured under a different term for Junos OS Evolved.

Figure 35 on page 327 is a sample illustration of inline monitoring services, where traffic is monitored at two different sampling rates on the device interface, and exported to four remote collectors in an IPFIX encapsulation format. For Junos OS, you configure the sampling rate on each collector, allowing different rates for each collector. For Junos OS Evolved, you configure the sampling rate on the inline-monitoring instance, and it applies to all of the collectors configured for that instance.

Figure 35: Inline Monitoring Services



In this example, the et-1/0/0 interface of the device is configured with inline monitoring services. The details of the configurations are as follows:

- There are two inline-monitoring instances — Instance 1 and Instance 2.
- There are four collectors, two collectors under each inline monitoring instance.
  - Instance 1 has Collector-1 and Collector-2.
  - Instance 2 has Collector-101 and Collector-102.
- The collectors on Instance 1 have a sampling rate of 1:10000.
- The collectors on Instance 2 have a sampling rate of 1:1.
- Instance 1 collectors have a source and destination address of 10.1.1.1 and 10.2.2.1, respectively.
- Instance 2 collectors have a source and destination address of 10.11.1.1 and 10.12.2.1, respectively.

- The packets are exported to the collectors in an IPFIX encapsulated format.

To configure inline monitoring services:

1. Define a firewall filter for each inline-monitoring instance for servicing the inline monitoring services. You can configure a family firewall filter with the term action `inline-monitoring-instance`.

   To define a firewall filter:

   ```
   [edit firewall family family filter filter-name term term]
   user@host# set from source-address source-IPv4-address
   user@host# set from destination-address destination-IPv4-address
   user@host# set then inline-monitoring-instance inline-monitoring-instance-name
   user@host# set then action
   ```

   In this example, Terms t1 and t2 are configured for Instance1 and Instance2, respectively.

   ```
   [edit firewall family inet filter SAMPLE_FOR_1 term t1]
   user@host# set from source-address 10.1.1.0/24
   user@host# set from destination-address 10.2.2.0/24
   user@host# set then inline-monitoring-instance Instance1
   user@host# set then accept
   user@host# set term t2 from source-address 10.11.1.0/24
   user@host# set term t2 from destination-address 10.12.2.0/24
   user@host# set term t2 then inline-monitoring-instance Instance2
   user@host# set term t2 then accept
   ```

2. Enable inline monitoring services by configuring the associated template, instance, and collector parameters.

   a. To configure the inline monitoring services template:

   ```
   [edit services inline-monitoring template template-name]
   user@host# set template-refresh-rate template-refresh-rate
   user@host# set option-template-refresh-rate option-template-refresh-rate
   user@host# set observation-domain-id observation-domain-id
   ```

   In this example, templates template-1 and template-2 are configured.

   ```
   [edit services inline-monitoring template template-1]
   user@host# set template-refresh-rate 60
   user@host# set option-template-refresh-rate 100
   ```

```
user@host# set observation-domain-id 1
[edit services inline-monitoring template template-2]
user@host# set template-refresh-rate 60
user@host# set option-template-refresh-rate 100
user@host# set observation-domain-id 2
```

b. To configure inline monitoring instance and collector parameters:

For Junos OS:

```
[edit services inline-monitoring instance inline-monitoring-instance-name]
user@host# set template-name template-name
user@host# set maximum-clip-length maximum-clip-length
user@host# set collector collector-name source-address source-IPv4-address
user@host# set collector collector-name destination-address destination-IPv4-address
user@host# set collector collector-name destination-port destination-port
user@host# set collector collector-name sampling-rate sampling-rate
```

In this example for Junos OS, Instance1 has two collectors, collector-1 and collector-2, and Instance2 has two collectors, collector-101 and collector-102. Different sampling rates have been configured for both the instances.

```
[edit services inline-monitoring instance Instance1]
user@host# set template-name template-1
user@host# set maximum-clip-length 126
user@host# set collector collector-1 source-address 10.1.1.1
user@host# set collector collector-1 destination-address 10.2.2.1
user@host# set collector collector-1 destination-port 2055
user@host# set collector collector-1 sampling-rate 10000
user@host# set collector collector-2 source-address 10.1.1.1
user@host# set collector collector-2 destination-address 10.2.2.1
user@host# set collector collector-2 destination-port 2055
user@host# set collector collector-2 sampling-rate 10000
```

```
[edit services inline-monitoring instance Instance2]
user@host# set template-name template-2
user@host# set maximum-clip-length 126
user@host# set collector collector-101 source-address 10.11.1.1
user@host# set collector collector-101 destination-address 10.12.2.1
user@host# set collector collector-101 destination-port 2055
```

```
user@host# set collector collector-101 sampling-rate 1
user@host# set collector collector-102 source-address 10.11.1.1
user@host# set collector collector-102 destination-address 10.12.2.1
user@host# set collector collector-102 destination-port 2055
user@host# set collector collector-102 sampling-rate 1
```

For Junos OS Evolved:

```
[edit services inline-monitoring instance inline-monitoring-instance-name]
user@host# set template-name template-name
user@host# set maximum-clip-length maximum-clip-length
user@host# set sampling-rate sampling-rate
user@host# set collector collector-name source-address source-IPv4-address
user@host# set collector collector-name destination-address destination-IPv4-address
user@host# set collector collector-name destination-port destination-port
```

In this example, for Junos OS Evolved, Instance1 has two collectors, collector-1 and collector-2, and Instance2 has two collectors, collector-101 and collector-102. Different sampling rates have been configured for both the instances.

```
[edit services inline-monitoring instance Instance1]
user@host# set template-name template-1
user@host# set maximum-clip-length 126
user@host# set sampling-rate 10000
user@host# set collector collector-1 source-address 10.1.1.1
user@host# set collector collector-1 destination-address 10.2.2.1
user@host# set collector collector-1 destination-port 2055
user@host# set collector collector-2 source-address 10.1.1.1
user@host# set collector collector-2 destination-address 10.2.2.1
user@host# set collector collector-2 destination-port 2055
```

```
[edit services inline-monitoring instance Instance2]
user@host# set template-name template-2
user@host# set maximum-clip-length 126
user@host# set sampling-rate 1
user@host# set collector collector-101 source-address 10.11.1.1
user@host# set collector collector-101 destination-address 10.12.2.1
user@host# set collector collector-101 destination-port 2055
user@host# set collector collector-102 source-address 10.11.1.1
```

```
user@host# set collector collector-102 destination-address 10.12.2.1
user@host# set collector collector-102 destination-port 2055
```

3. Map the firewall filter under the family of the logical unit of the interface to apply inline monitoring in the ingress or egress direction.

   Alternatively, you can apply inline monitoring by mapping the firewall filter to a forwarding table filter with an input or output statement to filter ingress or egress packets, respectively.

   To attach the firewall filter:

   ```
   [edit interfaces interface-name]
   user@host# set unit 0 family family filter input filter
   user@host# set unit 0 family family address ip-address
   ```

   In this example, the inline monitoring filter is attached to family inet of unit 0 of et-1/0/0.

   ```
   [edit interfaces et-1/0/0]
   user@host# set unit 0 family inet filter input SAMPLE_FOR_1
   user@host# set unit 0 family inet address 10.100.0.1/30
   ```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
23.4R1-EVO	Inline monitoring services (PTX10003 router with the JNP10K-LC1201 or JNP10K-LC1202 linecards)—Starting in Junos OS Evolved Release 23.4R1, you can configure inline monitoring services on the PTX10003 router to report exceptions. You can also configure the `any`, `ccc`, `ethernet-switch`, `inet`, `inet6`, or `mpls` family firewall filters with the term action inline-monitoring-instance *inline-monitoring-instance-name*.
22.4R1-EVO	Inline monitoring services (PTX10001-36MR, PTX10004, PTX10008, and PTX10016 routers with either the JNP10K-LC1201 or JNP10K-LC1202 linecards) - Starting in Junos OS Evolved Release 22.4R1, you can configure inline monitoring services on the PTX10001-36MR, PTX10004, PTX10008, and PTX10016 routers to sample packets, add metadata, and export the packets up to the configured clip length to an IPFIX collector for further processing. You can also configure the `any`, `ccc`, `ethernet-switch`, `inet`, `inet6`, or `mpls` family firewall filters with the term action inline-monitoring-instance *inline-monitoring-instance-name*.
22.3R1	Inline monitoring services (MX304 routers) - Starting in Junos OS Release 22.3R1, you can configure inline monitoring services on the MX304 router.

22.2R1-EVO	Inline monitoring services (PTX10001-36MR, PTX10004, PTX10008, and PTX10016 routers with either the JNP10K-LC1201 or JNP10K-LC1202 linecards) - Starting in Junos OS Evolved Release 22.1R1, you can configure inline monitoring services on the PTX10001-36MR, PTX10004, PTX10008, and PTX10016 routers to report exceptions. You can also configure the `any`, `ccc`, `ethernet-switch`, `inet`, `inet6`, or `mpls` family firewall filters with the term action inline-monitoring-instance *inline-monitoring-instance-name*.
21.4R1	Inline monitoring services (LC9600 linecard for the MX10008 router) - Starting in Junos OS Release 21.4R1, you can configure inline monitoring services on MX10008 routers that contain the LC9600 linecard.
21.2R1	Support for Layer 2 and any firewall filter families for inline monitoring services (MX Series with MPC10E and MPC11E linecards)—Starting in Junos OS Release 21.2R1, you can configure the `any`, `bridge`, `ccc`, `mpls`, or `vpls` family firewall filters with the term action inline-monitoring-instance *inline-monitoring-instance-name*.
21.2R1	Inline monitoring services (LC480 linecard for MX10008 and MX10016 routers - Starting in Junos OS Release 21.2R1, you can configure inline monitoring services on MX10008 and MX10016 routers that contain the LC480 linecard.
21.1R1	Support for Layer 2 and any firewall filter families for inline monitoring services (MX Series with MPCs excluding MPC10E and MPC11E linecards)—Starting in Junos OS Release 21.1R1, you can configure the any, bridge, ccc, mpls, or vpls family firewall filters with the term action inline-monitoring-instance *inline-monitoring-instance-name*.
20.4R1	Inline monitoring services (MPC10E and MPC11E linecards for MX Series routers - Starting in Junos OS Release 20.4R1, you can configure inline monitoring services on MX Series routers that contain the MPC10E and MPC11E linecards.
19.4R1	Inline monitoring services (MX Series with MPCs excluding MPC10E and MPC11E linecards) - Starting in Junos OS Release 19.4R1, you can configure a new monitoring technology that provides the flexibility to monitor different streams of traffic at different sampling rates on the same interface. You can also export the packet up to the configured clip length to a collector in an IP Flow Information Export (IPFIX) format. The IPFIX format includes important metadata information about the monitored packets for further processing at the collector.

CHAPTER 9

Flow-Based Telemetry

IN THIS CHAPTER

Flow-Based Telemetry (EX4100, EX4100-F, and EX4400 Series)

SUMMARY

Flow based telemetry (FBT) enables per-flow-level analytics, using inline monitoring services to create flows, collect them, and export them to a collector using the open standard IPFIX template to organize the flow.

IN THIS SECTION

FBT Overview

IN THIS SECTION

You can configure flow-based telemetry (FBT) for the EX4100, EX4100-F, and EX4400 Series switches. FBT enables per-flow-level analytics, using inline monitoring services to create flows, collect them, and export them to a collector. With inline monitoring services, you can monitor every IPv4 and IPv6 packet on both ingress and egress directions of an interface. A flow is a sequence of packets that have the same source IP, destination IP, source port, destination port, and protocol on an interface. For each flow, the software collects various parameters and exports the actual packet up to the configured clip length to a collector using the open standard IPFIX template to organize the flow. Once there is no active traffic for a flow, the flow is aged out after the configured inactive-timeout period (configure the `flow-inactive-timeout` statement at the [edit services inline-monitoring template *template-name*] hierarchy level). The software exports a IPFIX packet periodically at the configured flow-export timer interval. The observation domain identifier is used in the IPFIX packet to identify which line card sent the packet to the collector. Once set, the software derives a unique identifier for each line card based upon the system value set here.

Benefits of FBT

With FBT, you can:

- Count packet, TTL, and TCP window ranges
- Track and count Denial of Service (DoS) attacks
- Analyze the load distribution of ECMP groups/link aggregation groups (LAG) over the member IDs (EX4100 and EX4100-F only)
- Track traffic congestion (EX4100 and EX4100-F only)
- Gather information about multimedia flows (EX4100 and EX4100-F only)
- Gather information on why packets are dropped (EX4100 and EX4100-F only)

FBT Flow Export Overview

See Figure 36 on page 335 for a sample template, which shows the information element IDs, names, and sizes:

Figure 36: Sample FBT Information Element Template

```
[CINT] Output information elements (total 21):
        output info element ( 0): elem =  15(                    Reserved) and size =  2
        output info element ( 1): elem =  14(              FlowtrackerGroup) and size =  2
        output info element ( 2): elem =  15(                    Reserved) and size = 42
        output info element ( 3): elem =   5(                  L4DstPort) and size =  2
        output info element ( 4): elem =   4(                  L4SrcPort) and size =  2
        output info element ( 5): elem =   1(                    DstIPv4) and size =  4
        output info element ( 6): elem =   0(                    SrcIPv4) and size =  4
        output info element ( 7): elem =   6(                 IPProtocol) and size =  1
        output info element ( 8): elem =  15(                    Reserved) and size =  1
        output info element ( 9): elem =  40(           TimestampNewLearn) and size =  6
        output info element (10): elem =   7(                    PktCount) and size =  4
        output info element (11): elem =   4(                  L4SrcPort) and size =  2
        output info element (12): elem =  15(                    Reserved) and size = 20
        output info element (13): elem =  47(             FlowtrackerCheck) and size =  4
        output info element (14): elem =  84(  IngDropReasonGroupIdVector) and size =  2
        output info element (15): elem =  83(                     IngPort) and size =  1
        output info element (16): elem =  15(                    Reserved) and size =  9
        output info element (17): elem =  47(             FlowtrackerCheck) and size =  4
        output info element (18): elem =  90(  EgrDropReasonGroupIdVector) and size =  2
        output info element (19): elem =  54(                     EgrPort) and size =  1
        output info element (20): elem =  15(                    Reserved) and size =  9
```

jn-000303

Figure 37 on page 335 shows the format of a sample IPFIX data template for FBT:

Figure 37: Sample FBT IPFIX Data Template

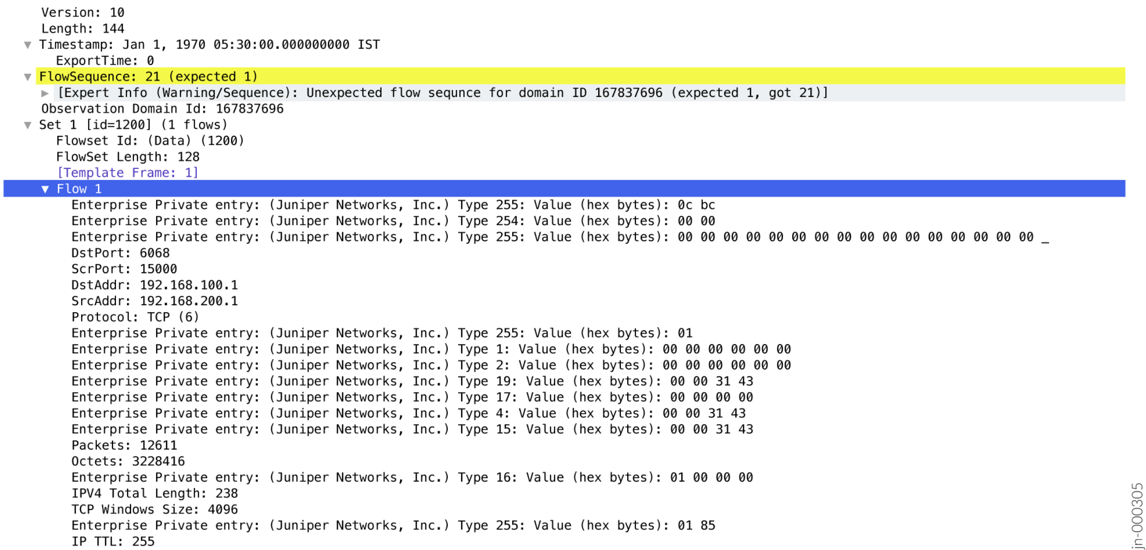
```
Apply a display filter ... <⌘/>
No.   Time        Source     Destination  Protoc  Length  Info
1269  180.070515  10.6.6.1   10.6.6.2     CFLOW   210  IPFIX flow (168 bytes)  Obs-Domain-ID=167837696  [Data-Template:1200]
1270  170.929160                          CFLOW   222  IPFIX flow (180 bytes)  Obs-Domain-ID=167837712  [Data-Template:1201]
1297  179.926594  10.6.6.1   10.6.6.2     CFLOW   186  IPFIX flow (144 bytes)  Obs-Domain-ID=167837696  [Data:1200]
1298  180.070515  10.6.6.1   10.6.6.2     CFLOW   210  IPFIX flow (168 bytes)  Obs-Domain-ID=167837696  [Data-Template:1200]
1299  180.070527  10.6.6.1   10.6.6.2     CFLOW   222  IPFIX flow (180 bytes)  Obs-Domain-ID=167837696  [Data-Template:1201]
1300  180.930733  10.6.6.1   10.6.6.2     CFLOW   210  IPFIX flow (168 bytes)  Obs-Domain-ID=167837712  [Data-Template:1200]

    ExportTime: 1603965361
  FlowSequence: 1
  Observation Domain Id: 167837712
▼ Set 1 [id=2] (Data Template): 1201
    FlowSet Id: (Data Template (V10 [IPFIX]) (2)
    FlowSet Length: 164
  ▼ Template (Id = 1201, Count = 24)
      Template Id: 1201
      Field Count: 24
    ▶ Field (1/24): 255 [pen: Juniper Networks, Inc.]
    ▶ Field (2/24): 254 [pen: Juniper Networks, Inc.]
    ▶ Field (3/24): 255 [pen: Juniper Networks, Inc.]
    ▶ Field (4/24): IPV6_SRC_ADDR
    ▶ Field (5/24): 255 [pen: Juniper Networks, Inc.]
    ▶ Field (6/24): L4_DST_PORT
    ▶ Field (7/24): L4_SRC_PORT
    ▶ Field (8/24): PROTOCOL
    ▶ Field (9/24): IPV6_DST_ADDR
    ▶ Field (10/24): 255 [pen: Juniper Networks, Inc.]
    ▶ Field (11/24):   1 [pen: Juniper Networks, Inc.]
    ▶ Field (12/24):   2 [pen: Juniper Networks, Inc.]
    ▶ Field (13/24):  19 [pen: Juniper Networks, Inc.]
    ▶ Field (14/24):  17 [pen: Juniper Networks, Inc.]
    ▶ Field (15/24):   4 [pen: Juniper Networks, Inc.]
    ▶ Field (16/24):  15 [pen: Juniper Networks, Inc.]
    ▶ Field (17/24): PKTS
    ▶ Field (18/24): BYTES
    ▶ Field (19/24):  16 [pen: Juniper Networks, Inc.]
    ▶ Field (20/24): payloadlengthIPv6
    ▶ Field (21/24): TCP_WINDOW_SIZE
    ▶ Field (22/24): 255 [pen: Juniper Networks, Inc.]
    ▶ Field (23/24): 254 [pen: Juniper Networks, Inc.]
    ▶ Field (24/24): 255 [pen: Juniper Networks, Inc.]
```

jn-000304

Figure 38 on page 336 shows the format of a sample exported IPFIX flow for FBT:

Figure 38: Sample Exported IPFIX Flow for FBT



```
Version: 10
Length: 144
▼ Timestamp: Jan 1, 1970 05:30:00.000000000 IST
    ExportTime: 0
▼ FlowSequence: 21 (expected 1)
  ▶ [Expert Info (Warning/Sequence): Unexpected flow sequnce for domain ID 167837696 (expected 1, got 21)]
  Observation Domain Id: 167837696
▼ Set 1 [id=1200] (1 flows)
    Flowset Id: (Data) (1200)
    FlowSet Length: 128
    [Template Frame: 1]
  ▼ Flow 1
      Enterprise Private entry: (Juniper Networks, Inc.) Type 255: Value (hex bytes): 0c bc
      Enterprise Private entry: (Juniper Networks, Inc.) Type 254: Value (hex bytes): 00 00
      Enterprise Private entry: (Juniper Networks, Inc.) Type 255: Value (hex bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 _
      DstPort: 6068
      ScrPort: 15000
      DstAddr: 192.168.100.1
      SrcAddr: 192.168.200.1
      Protocol: TCP (6)
      Enterprise Private entry: (Juniper Networks, Inc.) Type 255: Value (hex bytes): 01
      Enterprise Private entry: (Juniper Networks, Inc.) Type 1: Value (hex bytes): 00 00 00 00 00 00
      Enterprise Private entry: (Juniper Networks, Inc.) Type 2: Value (hex bytes): 00 00 00 00 00 00
      Enterprise Private entry: (Juniper Networks, Inc.) Type 19: Value (hex bytes): 00 00 31 43
      Enterprise Private entry: (Juniper Networks, Inc.) Type 17: Value (hex bytes): 00 00 00 00
      Enterprise Private entry: (Juniper Networks, Inc.) Type 4: Value (hex bytes): 00 00 31 43
      Enterprise Private entry: (Juniper Networks, Inc.) Type 15: Value (hex bytes): 00 00 31 43
      Packets: 12611
      Octets: 3228416
      Enterprise Private entry: (Juniper Networks, Inc.) Type 16: Value (hex bytes): 01 00 00 00
      IPV4 Total Length: 238
      TCP Windows Size: 4096
      Enterprise Private entry: (Juniper Networks, Inc.) Type 255: Value (hex bytes): 01 85
      IP TTL: 255
```

jn-000305

Table 51: Element Mapping

Element	Enterprise Element ID	Description
TIMESTAMP_FLOWSTART_VAL	1	Indicates the timestamp at which the TCP flow collection was started.
TIMESTAMP_FLOWEND_VAL	2	Indicates the timestamp at which the TCP flow collection was ended.
TIMESTAMP_NEW_LEARN_VAL	3	Timestamp when a new flow is learned in the flow table.
PKT_RANGE_CNTR1_VAL	4	Provides the number of packets in different size categories. User can opt for 4 categories or 6 categories under the template. The system categorises the packets into different size buckets accordingly and count. (counter-profile feature)

Table 51: Element Mapping *(Continued)*

Element	Enterprise Element ID	Description
PKT_RANGE_CNTR2_VAL	5	Provides the number of packets in different size categories. User can opt for 4 categories or 6 categories under the template. The system categorises the packets into different size buckets accordingly and count. (counter-profile feature)
PKT_RANGE_CNTR3_VAL	6	Provides the number of packets in different size categories. User can opt for 4 categories or 6 categories under the template. The system categorises the packets into different size buckets accordingly and count. (counter-profile feature)
PKT_RANGE_CNTR4_VAL	7	Provides the number of packets in different size categories. User can opt for 4 categories or 6 categories under the template. The system categorises the packets into different size buckets accordingly and count. (counter-profile feature)
PKT_RANGE_CNTR5_VAL	8	Provides the number of packets in different size categories. User can opt for 4 categories or 6 categories under the template. The system categorises the packets into different size buckets accordingly and count. (counter-profile feature)
PKT_RANGE_CNTR6_VAL	9	Provides the number of packets in different size categories. User can opt for 4 categories or 6 categories under the template. The system categorises the packets into different size buckets accordingly and count. (counter-profile feature)
PKT_RANGE_CNTR7_VAL	10	Provides the number of packets in different size categories. User can opt for 4 categories or 6 categories under the template. The system categorises the packets into different size buckets accordingly and count. (counter-profile feature)
PKT_RANGE_CNTR8_VAL	11	Provides the number of packets in different size categories. User can opt for 4 categories or 6 categories under the template. The system categorises the packets into different size buckets accordingly and count. (counter-profile feature)

Table 51: Element Mapping *(Continued)*

Element	Enterprise Element ID	Description
MIN_PKT_LENGTH_VAL	12	Provides the number of packets which have a size above the defined size. The configurable size range is between 64 and 9000 bytes.
MAX_PKT_LENGTH_VAL	13	Provides the number of packets which have a size below the defined size. The configurable size range is between 64 and 9000 bytes.
TCP_WINDOW_RANGE_CNTR_VAL	15	Counts the packets within the specified TCP window range.
DOS_ATTACK_ID_VAL	16	Reports the DDOS attack vector.
TTL_RANGE1_CNTR_VAL	17	Provides the number of packets within specific TTL value range.
TTL_RANGE2_CNTR_VAL	18	Provides the number of packets within specific TTL value range.
DOS_ATTACK_PKT_CNTR_VAL	19	Number of DDOS attack packets.
CUSTOM_PKT_RANGE_START_VAL	20	Provides the number of packets within the configured size range. You can define the size range between 64 and 9000 bytes by configuring the `counter-profile` statement at the `[edit services inline-monitoring]` hierarchy level. For example: `set services inline-monitoring counter-profile c1 counter p1 counter-type packet-range min-value 1000 max-value 1500`.
CUSTOM_TTL_RANGE_START_VAL	30	Provides the number of packets within the configured TTL range. You can define the TTL range between 0 and 255 by configuring the `counter-profile` statement at the `[edit services inline-monitoring]` hierarchy level. For example: `set services inline-monitoring counter-profile c1 counter p1 counter-type ttl-range min-value 10 max-value 15`.

Table 51: Element Mapping *(Continued)*

Element	Enterprise Element ID	Description
CUSTOM_TCP_WINDOW_RANGE_START_VAL	40	Provides the number of packets within the configured TCP window range. You can define the TCP window range between 0 and 65535 by configuring the `counter-profile` statement at the `[edit services inline-monitoring]` hierarchy level. For example: `set services inline-monitoring counter-profile c1 counter p1 counter-type tcp-window-range min-value 100 max-value 5000`.
INTER_ARRIVAL_TIME	50	The time difference between two consecutive packets at ingress (per flow).
INTER_DEPARTURE_TIME	51	The time difference between two consecutive packets at egress (per flow).
CHIP_DELAY	52	The amount of time the packet takes to transit the ASIC.
SHARED_POOL_CONGESTION	53	Shared pool congestion level
QUEUE_CONGESTION_LEVEL	54	Queue congestion level
INGRESS_DROP_REASON	55	The reason the packet is dropped at ingress.
INGRESS_DROP_REASON_PKT_CNTR_VAL	56	Number of packets dropped at ingress.
EGRESS_DROP_REASON	57	The reason the packet is dropped at egress.
EGRESS_DROP_REASON_PKT_CNTR_VAL	58	Number of packets dropped at egress.

Table 51: Element Mapping *(Continued)*

Element	Enterprise Element ID	Description
AGGREGATE_INTF_MEMBER_ID	59	ID for a member of a link aggregation group (LAG) or an equal-cost multipath (ECMP) group
AGGREGATE_INTF_GROUP_ID	60	ID for a link aggregation group (LAG)
MMU_QUEUE_ID	61	Indicates the queue ID to which the packet belongs.
UNKNOWN_ID_VAL	254	Not applicable to customer. Internal to Juniper.
RESERVED_ID_VAL	255	Not applicable to customer. Internal to Juniper.

When you create a new inline monitoring services configuration or change an existing one, the software immediately sends the periodic flow export of the data template to the respective collectors, instead of waiting until the next scheduled send time.

Limitations and Caveats

- IRB interfaces are supported.Starting in Junos OS Release 25.2R1 L2 firewall filters are supported.
- Only 8 inline-monitoring instances and 8 collectors per instance are supported.
- Flow records are limited to 128 bytes in length.
- The collector must be reachable through either the loopback interface or a network interface, not only through a management interface.
- You can configure a collector only within the same routing instance as the data. You cannot configure a collector within a different routing instance.
- You cannot configure an option template identifier or a forwarding class.
- The IPFIX Option Data Record and IPFIX Option Data Template are not supported.

- Feature profiles are not supported on EX4400 switches.
- If you make any changes to the feature-profile configuration, you must reboot the device.
- (EX4100 and EX4100-F only) If you configure any of the congestion or egress features in the feature profile for an inline-monitoring instance, you cannot configure a counter profile for a template in that instance.
- (EX4100 and EX4100-F only) Because the congestion and egress features collect a lot of data, you can only configure 4 or 5 of these features per inline-monitoring instance.
- (EX4100 and EX4100-F only) For multicast flow tracking, one ingress copy can produce multiple egress copies. All copies may update the same entry. Therefore, you can track the aggregate results of all copies of the same multicast flow.

Licenses

You must get a permanent license to enable FBT. To check if you have a license for FBT, issue the `show system license` command in operational mode:

```
user@host> show system license
License usage:
                                 Licenses     Licenses    Licenses    Expiry
  Feature name                       used    installed      needed
  Flow Based Telemetry                  1            1           0    permanent
Licenses installed:
  License identifier: XXXXXXXXXXXXXX
  License version: 4
  Order Type: commercial
  Valid for device: XXXXXXXXXXXX
  Features:
    Flow Based Telemetry - License for activating Flow Based Telemetry
      Permanent
```

For the EX4100 and EX4100-F switches, you need license S-EX4100-FBT-P. For the EX4400 switches, you need license S-EX-FBT-P.

Drop Vectors (EX4100 and EX4100-F only)

FBT can report more than 100 drop reasons. Drop vectors are very large vectors, too large to be reasonably accommodated in a flow record. Therefore, the software groups and compresses the drop vectors into a 16-bit compressed drop vector, and then passes that drop vector to the flow table. The 16-bit compressed drop vector corresponds to a particular drop vector group. Table 52 on page 342 and

Table 53 on page 342 describe how drop vectors are grouped together to form a particular 16-bit compressed drop vector.

Table 52: Ingress Drop Vector Groups (EX4100 and EX4100-F only)

Group ID	Drop Reason
1	MMU drop
2	TCAM, PVLAN
3	DoS attack or LAG loopback fail
4	Invalid VLAN ID, invalid TPID, or the port is not in the VLAN
5	Spanning Tree Protocol (STP) forwarding, bridge protocol data unit (BPDU), Protocol, CML
6	Source route, L2 source discard, L2 destination discard, L3 disable, and so on.
7	L3 TTL, L3 Header, L2 Header, L3 source lookup miss, L3 destination lookup miss
8	ECMP resolution, storm control, ingress multicast, ingress next-hop error

Table 53: Egress Drop Vector Groups (EX4100 and EX4100-F only)

Group ID	Drop Reason
1	MMU unicast traffic
2	MMU weighted random early detection (WRED) unicast traffic
3	MMU RQE

Table 53: Egress Drop Vector Groups (EX4100 and EX4100-F only) ***(Continued)***

Group ID	Drop Reason
4	MMU multicast traffic
5	Egress TTL, stgblock
6	Egress field processor drops
7	IPMC drops
8	Egress quality of service (QoS) control drops

Configure FBT (EX4100, EX4100-F, and EX4400 Series)

FBT enables per-flow-level analytics, using inline monitoring services to create flows, collect them, and export them to a collector. A flow is a sequence of packets that have the same source IP, destination IP, source port, destination port, and protocol on an interface. For each flow, various parameters are collected and sent to a collector using the open standard IPFIX template to organize the flow. Once there is no active traffic for a flow, the flow is aged out after the configured inactive-timeout period (configure the `flow-inactive-timeout` statement at the [edit services inline-monitoring template *template-name*] hierarchy level). The software exports a IPFIX packet periodically at the configured flow-export timer interval. The observation domain identifier is used in the IPFIX packet to identify which line card sent the packet to the collector. Once set, the software derives a unique identifier for each line card based upon the system value set here.

To configure flow-based telemetry:

1. Define the IPFIX template.

   To configure attributes of the template:

   ```
   user@host# set services inline-monitoring template template_1 flow-inactive-timeout seconds
   user@host# set services inline-monitoring template template_1 observation-domain-id identifier
   user@host# set services inline-monitoring template template_1 template-refresh-rate template-
   refresh-rate
   user@host# set services inline-monitoring template template_1 template-id template-id
   ```

In this example, the inactive-flow timeout period is set to 10 seconds, the observation domain ID is set to 25, the template refresh rate is set to 30 seconds, and you've configured a template identifier

```
user@host# set services inline-monitoring template template_1 flow-inactive-timeout 10
user@host# set services inline-monitoring template template_1 observation-domain-id 25
user@host# set services inline-monitoring template template_1 template-refresh-rate 30
user@host# set services inline-monitoring template template_1 template-id 32768
```

2. Attach a template to the instance and describe the collector.

   To configure the instance and collector:

```
user@host# set services inline-monitoring instance instance-name template-name template-name
user@host# set services inline-monitoring instance instance-name collector collector-name
source-address IPv4-address
user@host# set services inline-monitoring instance instance-name collector collector-name
destination-address IPv4-address
user@host# set services inline-monitoring instance instance-name collector collector-name
dscp dscp-bits
user@host# set services inline-monitoring instance instance-name collector collector-name
destination-port port
```

   In this example, you create a template with the name `template_1`, create an inline-monitoring instance `i1`, and create the configuration for the collector `c2`:

```
user@host# set services inline-monitoring instance i1 template-name template_1
user@host# set services inline-monitoring instance i1 collector c2 source-address 10.11.12.1
user@host# set services inline-monitoring instance i1 collector c2 destination-address
10.11.12.2
user@host# set services inline-monitoring instance i1 collector c2 dscp 21
user@host# set services inline-monitoring instance i1 collector c2 destination-port 2055
```

3. Create a firewall filter and configure the action `inline-monitoring-instance`.

   To configure the firewall filter:

```
user@host# set firewall family inet filter filter-name term term-name from source-address
source-address
user@host# set firewall family inet filter filter-name term term-name from destination-
address destination-address
```

```
user@host# set firewall family inet filter filter-name term term-name then inline-monitoring-
instance instance-name
user@host# set firewall family inet filter filter-name term term-name then accept
```

In this example, you configure an IPv4 firewall filter named `ipv4_ingress`, with the term name `rule1` containing the action `inline-monitoring-instance`, and the inline monitoring instance `i1` is mapped to it:

```
user@host# set firewall family inet filter ipv4_ingress term rule1 from source-address
10.11.12.1
user@host# set firewall family inet filter ipv4_ingress term rule1 from destination-address
10.11.12.2
user@host# set firewall family inet filter ipv4_ingress term rule1 then inline-monitoring-
instance i1
user@host# set firewall family inet filter ipv4_ingress term rule1 then accept
```

4. Map the firewall filter to the family under the logical unit of the already-configured interface to apply inline monitoring in the ingress direction.

   To map the firewall filter:

```
user@host# set interface interface-name unit 0 family inet filter input filter-name
```

   In this example, you map the `ipv4_ingress` firewall filter to the `inet` family of logical interface 0 of the physical interface et-0/0/1:

```
user@host# set interface et-0/0/1 unit 0 family inet filter input ipv4_ingress
```

5. (Optional) Configure the sampling profile and rate, configure the profile for which counters to export to the collector, configure the flow rate and burst size, and enable security analytics for flow-based telemetry:

   To configure the flow-monitoring properties:

```
user@host# set services inline-monitoring template template _1 flow-monitoring sampling-
profile profile-name
user@host# set services inline-monitoring template template _1 flow-monitoring sampling-rate
rate
user@host# set services inline-monitoring template template _1 flow-monitoring counter-
profile profile-identifier
user@host# set services inline-monitoring template template _1 flow-monitoring flow-rate kbps
```

```
burst-size bytes
user@host# set services inline-monitoring template template _1 flow-monitoring security-enable
```

In this example, the sampling profile is set to Random, the sampling rate is set to every 512 bytes, the counter profile is set to Per_flow_6_counters, the flow-rate is set to 100000 kbps, the burst-size is set to 2048 bytes, and security analytics are enabled:

```
user@host# set services inline-monitoring template template _1 flow-monitoring sampling-
profile Random
user@host# set services inline-monitoring template template _1 flow-monitoring sampling-rate
512
user@host# set services inline-monitoring template template _1 flow-monitoring counter-
profile Per_flow_6_counters
user@host# set services inline-monitoring template template _1 flow-monitoring flow-rate
100000 burst-size 2048
user@host# set services inline-monitoring template template _1 flow-monitoring security-enable
```

6. (Optional, EX4100 and EX4100-F switches only) Configure a feature profile to collect more data about packets as they move through the switch.

   For example, you could monitor congestion or collect information about why packets are being dropped. You can enable security analytics either here or in the previous step. To configure a feature profile:

```
user@host# set services inline-monitoring feature-profile feature_1 features aggregate-intf-
member-id
user@host# set services inline-monitoring feature-profile feature_1 features chip-delay
user@host# set services inline-monitoring feature-profile feature_1 features egress-drop-
reason
user@host# set services inline-monitoring feature-profile feature_1 features flow-start-end-
time
user@host# set services inline-monitoring feature-profile feature_1 features ingress-drop-
reason
user@host# set services inline-monitoring feature-profile feature_1 features inter-arrival-
time
user@host# set services inline-monitoring feature-profile feature_1 features inter-departure-
time
user@host# set services inline-monitoring feature-profile feature_1 features queue-congestion-
level
user@host# set services inline-monitoring feature-profile feature_1 features security-enable
user@host# set services inline-monitoring feature-profile feature_1 features shared-pool-
congestion
```

You must reboot the system for the feature profile to take effect. Because the aggregate interface distribution monitoring, congestion, and egress features collect a lot of data, you can only configure 4 or 5 of these features per inline-monitoring instance. The statements that configure these features are:

- `aggregate-intf-member-id`
- `egress-drop-reason`
- `inter-departure-time`
- `queue-congestion-level`
- `shared-pool-congestion`

After you commit the configuration and reboot the system, use the `show services inline-monitoring feature-profile-mapping fpc-slot` *`slot-number`* command to verify that the features have been successfully configured.

7. After committing the configuration, monitor inline-monitoring statistics with the `show services inline-monitoring statistics fpc-slot` *`slot-number`* command.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
22.2R1	You can now configure flow-based telemetry (FBT) for the EX4100 and EX4100-F Series switches, and configure additional items to track for a flow using the `feature-profile` *`name`* `features` statement at the `[edit inline-monitoring]` hierarchy level.
21.1R1	You can configure flow-based telemetry (FBT) for the EX4400 Series switches. FBT enables per-flow-level analytics, using inline monitoring services to create flows, collect them, and export them to a collector.

RELATED DOCUMENTATION

Inline Monitoring Services Configuration | 318

inline-monitoring

flow-monitoring (Inline Monitoring Services)

features

Flow-Based Telemetry for VXLANs (QFX5120)

SUMMARY

Flow based telemetry (FBT) for VXLANs in Junos OS enables per-flow-level analytics on IRB interfaces, using inline monitoring services to create flows, collect them, and export them to a collector using the open standard IP Flow Information Export (IPFIX) template to organize the flow.

IN THIS SECTION

FBT for VXLANs Overview

IN THIS SECTION

You can configure flow-based telemetry (FBT) for VXLANs for the QFX5120 -32C and QFX5120-48y-8c switches. FBT for VXLANs enables inline telemetry data for VXLANs that have either centrally-routed bridging (CRB) or edge-routed bridging (ERB) overlays. FBT for VXLANs enables per-flow-level analytics for IRB interfaces, using inline monitoring services to create flows, collect them, and export them to a collector. With inline monitoring services, you can monitor every IPv4 and IPv6 packet on both ingress and egress directions of an interface.

Benefit of FBT for VXLANs

With FBT for VXLANs, you can enable inline telemetry data for EVPN-VXLAN architectures that have either CRB or ERB overlays, giving you an additional source of information about your network,

A VXLAN with a CRB overlay has core switches configured as Layer/Layer 3 VXLAN gateways where the Integrated Routing and Bridging (IRB) interfaces for the virtual networks are configured on the core switches. In contrast, core switches in a VXLAN with an ERB overlay provide transport of EVPN type-2 and type-5 routes and the IRB interfaces are configured on the distribution switches. The ERB design also enables faster server-to-server, intra-campus traffic. As a result, with an ERB overlay, routing happens much closer to the end systems than with a CRB overlay. Figure 39 on page 349and Figure 40

on page 350 show sample topologies for these overlays. To learn more about these EVPN-VXLAN architectures, see Technology Primer: EVPN-VXLAN Fabrics for the Campus.

Figure 39: Centrally-Routed Bridging (CRB) Topology

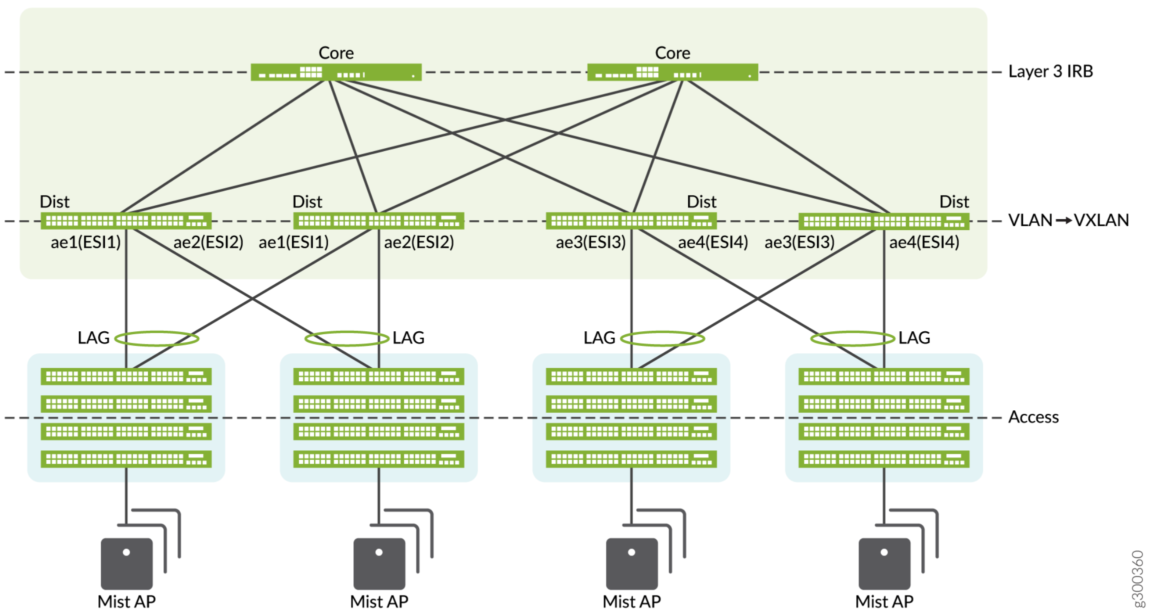
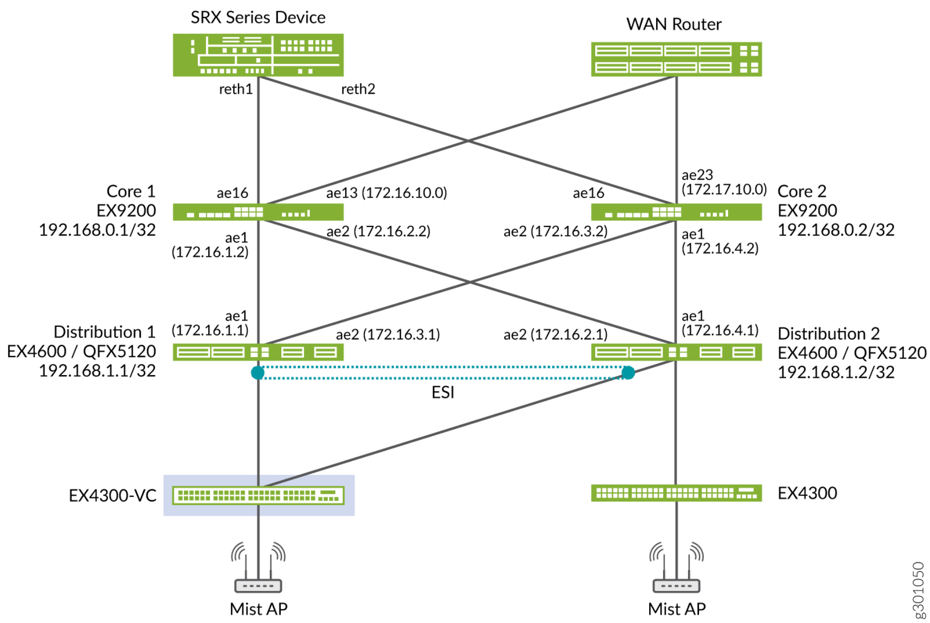


Figure 40: Edge-Routed Bridging (ERB) Topology



Flow Export Overview

FBT for VXLANs uses software-based IPFIX flow export. (IPFIX is defined in RFC 7011.) A flow is a sequence of packets that have the same core set of parameters on an interface, some of which are source IP, destination IP, source port, destination port, and protocol. This core set of parameters is called a flow key, and the software uses this key to learn about the flows. For each flow, the software collects various parameters and exports the actual packet up to the configured clip length to a collector using the open standard IPFIX template to organize the flow. Once there is no active traffic for a flow, the flow is aged out after the configured inactive-timeout period (configure the `flow-inactive-timeout` statement at the `[edit services inline-monitoring template` *`template-name`*`]` hierarchy level).

For FBT for VXLANs, the flow key differs depending on whether you are monitoring IPv4 or IPv6 traffic. The flow key for IPv4 traffic is explained in Table 54 on page 351 and the flow key for IPv6 traffic is explained in Table 55 on page 351. For both IPv4 and IPv6 traffic, in addition to the key fields, the flow contains fields for the ingress and egress ports, the flow start and end time, and the byte and packet count delta. The flow start time is the timestamp for when the software learned the flow. The flow stop time is the timestamp of the latest counter query. A sample IPFIX data template for IPv4 traffic is shown in Figure 41 on page 352.

Table 54: IPv4 Flow Key

Field	Field size in bytes
Source IP address	4
Destination IP address	4
Protocol (TCP or UDP)	1
Source port (TCP or UDP)	2
Destination port (TCP or UDP)	2
Virtual routing and forwarding table (VRF) identifier	2
Ingress port	1
VXLAN network identifier (layer 2 segment ID)	3

Table 55: IPv6 Flow Key

Field	Field size in bytes
Source IP address	4
Destination IP address	4
Protocol (TCP or UDP)	1
Source port (TCP or UDP)	2
Destination port (TCP or UDP)	2
Virtual routing and forwarding table (VRF) identifier	2

Figure 41: Sample IPFIX Data Template for IPv4 Traffic

```
  Version: 10
  Length: 76
▼ Timestamp: Jan 24, 2022 06:18:29.000000000 Pacific Standard Time
  FlowSequence: 1159
  Observation Domain Id: 167837696
▼ Set 1 [id=2] (Data Template): 65000
    Flowset Id: Data Template (V10 [IPFIX] (2)
    FlowSet Length: 60
  ▼ Template (Id = 65000, Count = 13)
        Template Id: 65000
        Field Count: 13
  ▼ Field (1/13): IP_SRC_ADDR
  ▼ Field (2/13): IP_DST_ADDR
  ▼ Field (3/13): PROTOCOL
  ▼ Field (4/13): L4_SRC_PORT
  ▼ Field (5/13): L4_DST_PORT
  ▼ Field (6/13): ingressVRFID
  ▼ Field (7/13): layer2SegmentId
  ▼ Field (8/13): INPUT_SNMP
  ▼ Field (9/13): OUTPUT_SNMP
  ▼ Field (10/13): flowStartSeconds
  ▼ Field (11/13): flowEndSeconds
  ▼ Field (12/13): PKTS
  ▼ Field (13/13): BYTES
```

jn-000342

Limitations and Caveats

- FBT for VXLANs is supported only on Junos OS.
- Only IRB interfaces are supported. For EVPN-VXLAN networks with CRB overlays, you can only monitor the IRB interfaces on the spine. For EVPN-VXLAN networks with ERB overlays, you can only monitor the IRB interfaces on the leaves.
- Only one inline-monitoring instance and one collector are supported.
- The collector must be reachable through a network interface, not only through a management or loopback interface.
- You cannot configure an option template identifier or a forwarding class.
- The IPFIX Option Data Record and IPFIX Option Data Template are not supported.
- Flow learning and tracking is based on client traffic data only, not the outer tunnel header. Flow learning is software-based and takes up to 10 seconds per flow.
- Counters are not active until the software learns the flow and installs the flow in the flow table.
- The software does not use the TCP FIN/RST flag for flow aging.
- The software requires a layer 3 header in the packet, and supports only the TCP and UDP protocols.

- The reported egress port might not be correct with LAG, ECMP, broadcast, multicast, or unknown traffic, if the egress port is in a different VRF.

Configure FBT for VXLANs (QFX5120)

You can configure flow-based telemetry (FBT) for VXLANs for the QFX5120 -32C and QFX5120-48y-8c switches. FBT for VXLANs enables inline telemetry data for VXLANs that have either centrally-routed bridging (CRB) or edge-routed bridging (ERB) overlays. FBT for VXLANs enables per-flow-level analytics for IRB interfaces, using inline monitoring services to create flows, collect them, and export them to a collector. With inline monitoring services, you can monitor every IPv4 and IPv6 packet on both ingress and egress directions of an interface.

Before you can configure FBT for VXLANs, you must first enable software-based IPFIX flow export and must allocate exact-match memory in the unified forwarding table to learn the flows. To configure:

```
user@host# set system packet-forwarding-options ipfix-sw-mode
user@host# set chassis forwarding-options em-hw-profile
user@host# commit
```

After you commit the configuration, the system then prompts you to reboot the system.

To configure FBT for VXLANs:

1. Define the IPFIX template.

   To configure attributes of the template:

   ```
   user@host# set services inline-monitoring template template_1 flow-inactive-timeout seconds
   user@host# set services inline-monitoring template template_1 template-refresh-rate template-
   refresh-rate
   user@host# set services inline-monitoring template template_1 template-identifier template-
   identifier
   user@host# set services inline-monitoring template template_1 template-type (ipv4-template |
   ipv6-template)
   ```

   In this example, the inactive-flow timeout period is set to 10 seconds, the template refresh rate is set to 30 seconds, you've configured a template identifier, and you're using the IPv4 template:

   ```
   user@host# set services inline-monitoring template template_1 flow-inactive-timeout 10
   user@host# set services inline-monitoring template template_1 template-refresh-rate 10
   ```

```
user@host# set services inline-monitoring template template_1 template-identifier 1200
user@host# set services inline-monitoring template template_1 template-type ipv4-template
```

2. Attach a template to the instance and describe the collector.

   FBT for VXLANs only supports IPv4 addresses for the collector. To configure the instance and collector:

```
user@host# set services inline-monitoring instance instance-name template-name template-name
user@host# set services inline-monitoring instance instance-name collector collector-name
source-address (IPv4-address)
user@host# set services inline-monitoring instance instance-name collector collector-name
destination-address (IPv4-address)
user@host# set services inline-monitoring instance instance-name collector collector-name
dscp dscp-bits
user@host# set services inline-monitoring instance instance-name collector collector-name
destination-port port
```

   In this example, you create a template with the name `template_1`, create an inline-monitoring instance `i1`, and create the configuration for the collector `c2` using IPv4 addresses:

```
user@host# set services inline-monitoring instance i1 template-name template_1
user@host# set services inline-monitoring instance i1 collector c2 source-address 10.11.12.1
user@host# set services inline-monitoring instance i1 collector c2 destination-address
10.11.12.2
user@host# set services inline-monitoring instance i1 collector c2 dscp 21
user@host# set services inline-monitoring instance i1 collector c2 destination-port 2055
```

3. Create a firewall filter and configure the action `inline-monitoring-instance`.

   To configure the firewall filter:

```
user@host# set firewall family inet filter filter-name term term-name from source-address
(IPv4-source-address | IPv6-source-address)
user@host# set firewall family inet filter filter-name term term-name from destination-
address (IPv4-destination-address | IPv6-destination-address)
user@host# set firewall family inet filter filter-name term term-name then inline-monitoring-
instance instance-name
user@host# set firewall family inet filter filter-name term term-name then accept
```

In this example, you configure an IPv4 firewall filter named `ipv4_ingress`, with the term name `rule1` containing the action `inline-monitoring-instance`, and the inline monitoring instance `i1` is mapped to it:

```
user@host# set firewall family inet filter ipv4_ingress term rule1 from source-address
10.11.12.1
user@host# set firewall family inet filter ipv4_ingress term rule1 from destination-address
10.11.12.2
user@host# set firewall family inet filter ipv4_ingress term rule1 then inline-monitoring-
instance i1
user@host# set firewall family inet filter ipv4_ingress term rule1 then accept
```

4. Map the firewall filter to the family under the logical unit of the already-configured interface to apply inline monitoring in the ingress direction.

   To map the firewall filter:

```
user@host# set interfaces irb unit unit-number family inet filter input filter-name
```

   In this example, you map the `ipv4_ingress` firewall filter to the `inet` family of unit 100:

```
user@host# set interface irb unit 100 family inet filter input ipv4_ingress
```

5. Commit the configuration.
6. Monitor inline-monitoring statistics with the `show services inline-monitoring statistics fpc-slot` *`slot-number`* command.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
22.2R1	You can configure flow-based telemetry (FBT) for VXLANs for the QFX5120 -32C and QFX5120-48y-8c switches. FBT for VXLANs enables inline telemetry data for VXLANs that have either centrally-routed bridging (CRB) or edge-routed bridging (ERB) overlays. FBT for VXLANs enables per-flow-level analytics for IRB interfaces, using inline monitoring services to create flows, collect them, and export them to a collector.

RELATED DOCUMENTATION

Inline Monitoring Services Configuration | 318

CHAPTER 10

Inband Flow Analyzer 2.0

IN THIS CHAPTER

Inband Flow Analyzer (IFA) 2.0 Probe for Real-Time Flow Monitoring

SUMMARY

Inband Flow Analyzer (IFA) 2.0 collects data on a per-hop basis across the network. You export this data to external collectors to perform localized or end-to-end analytics.

IN THIS SECTION

Inband Flow Analyzer 2.0

IN THIS SECTION

Inband Flow Analyzer 2.0 Overview

Inband Flow Analyzer 2.0 (IFA 2.0) is a feature that you can use to monitor and analyze packets as they enter and exit the network. As the network administrator, you can use this feature to collect data related to the paths the packets take through the network and how long the packets spend at each hop. This data provides an indication of excessive latency and possible congestion. This feature helps you to get insights about complex networks by collecting per-hop flow data on the data plane.

IFA uses probe packets to collect network-wide flow data. IFA samples the flow of interest and generates probe packets. These packets are representative of the original flow, possessing the same characteristics as the original flow. This means that IFA packets traverse the same path in the network and the same queues in the networking element as the original packet would. As a result, IFA probe packets traverse the same network path as the original flow, experiencing similar latency and congestion.

You can use Inband Flow Analyzer 2.0 (IFA 2.0) to collect flow data information such as:

- Residence time (latency)
- Per-hop latency
- Per-hop ingress port number
- Per-hop egress port number
- Received packet timestamp value
- Queue ID
- Congestion notification
- Egress port speed

IFA 2.0 is defined in the IETF draft titled Inband Flow Analyzer, draft-kumar-ippm-ifa-02.

Benefits

- IFA probe packets traverse the same network path as the original flow, helping you to monitor the network for faults and performance issues.

- Monitors live traffic and thus helps to perform packet-level latency analysis and queue-congestion monitoring to optimize the network performance.

Modes (Junos OS Evolved only)

Starting in Junos OS Evolved Release 25.4R1, we support two modes for the IFA 2.0 feature. The new modes are as follows:

- In live mode, for those switches that support it, IFA 2.0 uses the original packets as IFA probe packets.
- In probe mode, IFA 2.0 uses copies of the original packets as the IFA probe packets. This mode is the default, and behaves the same way as this feature did prior to Junos OS Evolved Release 25.4R1.

When you configure an IFA initiator node to initiate IFA probe packets:

- In live mode, the initiator inserts IFA headers and metadata into the live packets.
- In probe mode, the initiator samples the flow and copies that sample to create the probe.

When you configure an IFA terminator node to parse the IFA header in the packet to determine if it is a live or probe packet:

- If it is a live packet, then the terminator strips the IFA layer and forwards the modified live packet to its destination.
- If the packet is a probe packet, then the terminator drops the packet and sends a mirror copy to the collector.

To specify the mode, configure the `mode (live | probe)` statement at the `[edit services inband-flow-telemetry]` hierarchy level. The default mode is `probe`. Use the `show services inband-flow-telemetry global` command to see probe results that include the mode information.

Inband Flow Analyzer Process

IFA uses the following processing nodes (as shown in Figure 42 on page 359) to monitor and analyze flows:

- IFA initiator node (also known as ingress node)
- IFA transit node
- IFA terminating node (also known as egress node)

IFA 2.0 supports processing both Layer 3 (L3) and VXLAN flows, but you can't configure IFA for both L3 and VXLAN flows on the same device. The flow-type options are mutually exclusive. You use the `flow-type` configuration statement to set the flow type of interest —either L3 or VXLAN. You configure the `flow-type` statement only for the IFA initiator and IFA terminating nodes (generally leaf nodes). For an IFA transit node (generally a spine node), you don't need to configure the `flow-type` statement.

Figure 42: IFA Processing

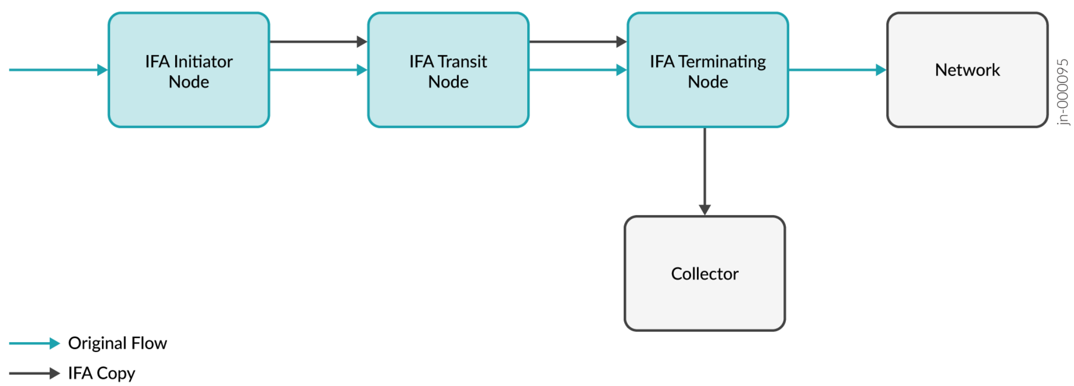


Table 56 on page 359 summarizes the different functions that the IFA processing nodes perform:

Table 56: IFA Node Functions

IFA Node	Function
IFA initiator node	Samples the flow traffic of interest (L3 or VXLAN) and creates an IFA copy by adding an IFA header to each sample. If you configured live mode, the initiator inserts IFA headers and metadata into the live packets.

Table 56: IFA Node Functions *(Continued)*

IFA Node	Function
IFA transit node	Identifies IFA packets and appends their metadata to the metadata stack in the packet. • If any packet comes with an IFA header, the node inserts the metadata into the metadata stack and forwards it. If the hop limit is 0, the node does not insert the metadata. • When a non-IFA device receives an IFA packet, the device forwards it without IFA processing. • The QFX5220 as an IFA transit node can not insert metadata into the metadata stack of the IFA probe packet header. Instead, the QFX5220 adds a tailstamp to the end of the IFA probe packet that includes timestamps and other metadata.
IFA terminating node	• Inserts terminating node metadata into an IFA packet. If you configured live mode, the packet is a live packet. The terminator strips the IFA layer and forwards the modified live packet to its destination. • Formats the IFA packets in IP Flow Information Export (IPFIX) format and sends the packets to the configured collector. You can use any collector (or application) that supports the IPFIX format. The QFX5240 switch does not use the IPFIX templates; instead, it uses the IPFIX PSAMP format to format the data sent to the collector. **NOTE**: IFA terminating functionality requires a valid Juniper Advanced Telemetry Feature (ATF) license.

IFA Probe Packet Headers

An IFA 2.0 probe packet contains the following:

- IFA Header
- IFA Metadata Header
- IFA Metadata Stack

Figure 43 on page 361 shows the L3 IFA 2.0 packet format at the IFA initiator node:

Figure 43: Layer 3 IFA 2.0 Packet Format at the IFA Initiator Node

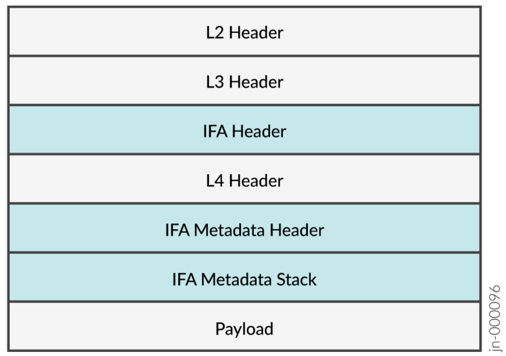
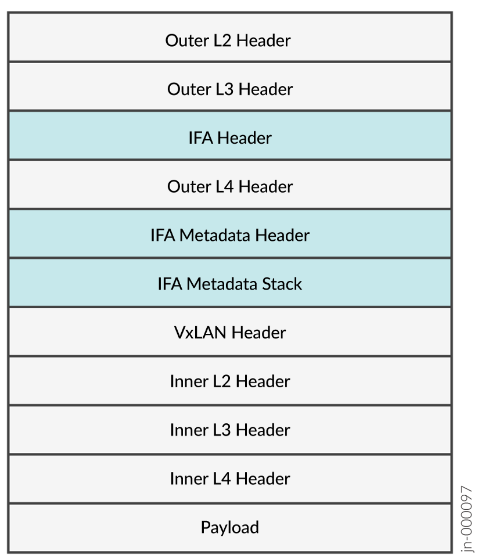


Figure 44 on page 361 shows the VXLAN IFA 2.0 packet format at the IFA initiator node.

Figure 44: VXLAN IFA 2.0 Packet Format at the IFA Initiator Node



NOTE: When VXLAN is used, then the IFA headers are added after VXLAN encapsulation using a three-pass mechanism.

IFA Header

IFA 2.0 defines an upper layer header (ULH), similar to how TCP, UDP, Generic Routing Encapsulation (GRE), and Spanning Tree Protocol (STP) define a ULH. The IFA ULH is always the first header after the IP header, even if there are some other IPv4 extension headers. The `NextHdr` field (that is, the `Protocol Type` field in the IFA header) carries the original IP header protocol field value. Figure 45 on page 362 shows the IFA header format.

Figure 45: IFA Header

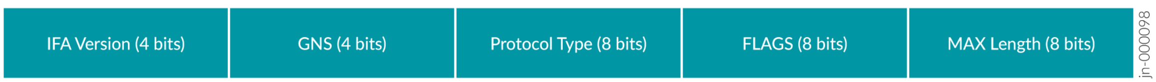


Table 57 on page 362 provides details about the IFA header fields.
Table 57: IFA Header Fields

IFA Header Field	Description
IFA Version	Version of the IFA header. In the current implementation, the IFA version is 2.0.
GNS	Global namespace (GNS) for IFA metadata. The IFA initiator node sets the value for this field as 0xF.
Protocol Type	IP header protocol type. This value is copied from the IP header.
FLAGS	Unused
MAX Length	Maximum allowed length of the metadata stack in multiples of four octets. The initiator node initializes this field. Each node in the path compares the current length with the maximum length. If the current length equals or exceeds the maximum length, the transit node stops inserting metadata. You can configure this maximum allowed length. The default value is 240 octets (for 30 hops).

IFA Metadata Header

IFA 2.0 defines a compact 4-byte metadata header as shown in Figure 46 on page 363. The IFA initiator node adds this header to the probe packet.

Figure 46: IFA Metadata Header Format

Table 58 on page 363 provides details about the IFA metadata header fields.

Table 58: IFA Metadata Header Fields

IFA Metadata Header Field	Description
Request Vector	Specifies the presence of fields as specified by the GNS. Unused.
Action Vector	Specifies the node-local or the end-to-end action on the IFA packets. Unused.
Hop Limit	Specifies the maximum number of allowed hops in an IFA zone. The initiator node initializes this field. The hop limit is decremented at each hop. If the hop limit of the incoming packet is 0, the current node does not insert metadata. You can configure this limit. The default value is 250. The terminating node does not perform the hop limit check.
Current Length	Specifies the current length of the metadata stack in multiples of 4 octets.

IFA Metadata Stack

Each IFA hop inserts hop-specific metadata into an IFA metadata stack as shown in Figure 47 on page 364. The IFA initiator node adds the metadata header after the L4 header.

The QFX5220 as a transit node can not insert metadata into the metadata stack of the IFA probe packet header. Instead, the QFX5220 adds a tailstamp to the end of the IFA probe packet that includes timestamps and other metadata. For more information about these tailstamps, see "Tailstamps for IFA Probe Packets (QFX5220 only)" on page 366.

The Egress Port Speed field in the IFA 2.0 Metadata stack is not supported on the QFX5230. The value in the field is always '0' irrespective of the egress port speed.

Figure 47: IFA Metadata Stack Header

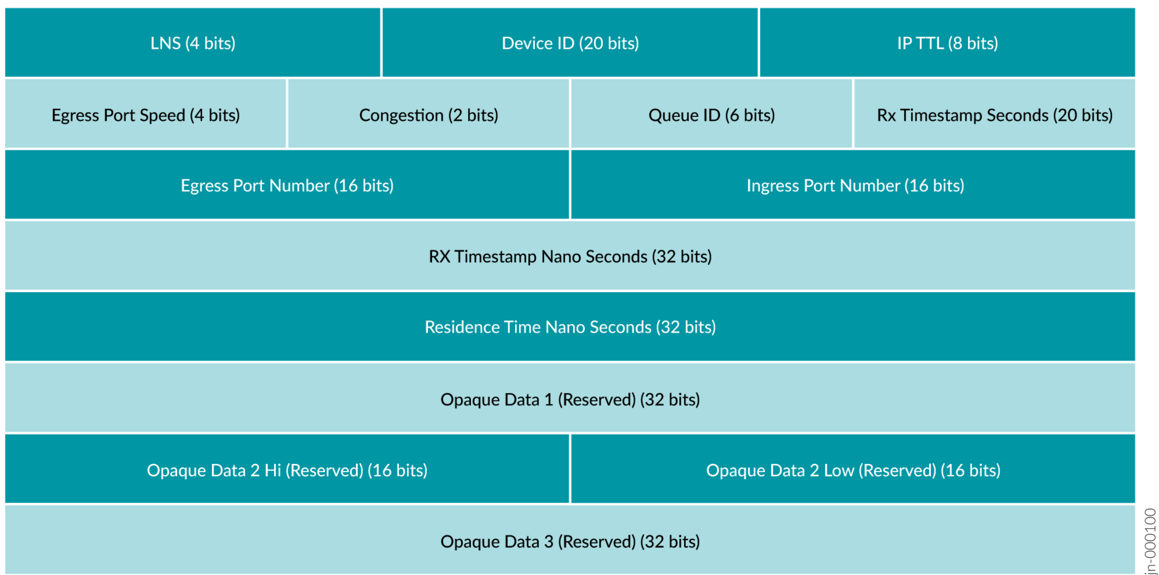


Table 59 on page 364 provides details about the IFA metadata stack header fields.

Table 59: IFA Metadata Stack Header Fields

IFA Metadata Stack Header Field	Description
LNS	Local namespace. You must set the LNS value to 1.
Device ID	User-configurable device ID. You can explicitly configure the device ID or configure the `device-id auto` statement. If you configure `device-id auto`, the device ID is internally generated from the router ID or the management IP address.
IP TTL	IP time-to-live (TTL) value at each hop.

Table 59: IFA Metadata Stack Header Fields *(Continued)*

IFA Metadata Stack Header Field	Description
Egress Port Speed	Encodings are 0–10Gbps, 1–25Gbps, 2–40Gbps, 3–50Gbps, 4–100Gbps, 5–200Gbps, 6–400Gbps. Egress port speed is mapped with IFA metadata. For example, when a egress port speed is 10Gbps, then the speed field of IFA packet is set to 0.
Congestion	Indicates whether the packet has experienced congestion. You must enable an explicit congestion notification (ECN) on the egress port.
Queue ID	Egress port queue ID.
Rx Timestamp Seconds	Received packet timestamp value (in seconds). It is the collector's responsibility to retrieve time-of-day (ToD) from these 20-bit values. 20-bit seconds will wrap around every 12 days. Collector has to periodically sync up ToD within the wraparound time and use it along with 20-bit from metadata to derive the 32-bit `Rx Timestamp Seconds` value.
Egress Port Number	Egress hardware (ASIC) port number.
Ingress Port Number	Ingress hardware port number.
Rx Timestamp Nano Seconds	Received timestamp value in nanoseconds. This timestamp is the exact value after the timestamp in seconds. Add this to the Rx Timestamp Seconds value to get the exact timestamp with nanoseconds precision.
Residence Time Nano Seconds	Per-hop latency in nanoseconds. For the QFX5120, the residence time is calculated as 0x3B9ACA00 (1 second in nanoseconds) + TX_NSEC - RX_NSEC. (An extra second is added to every packet to avoid wraparound handling.) In contrast, for the QFX5130, QFX5220, and QFX5700, the residence time is updated as the actual value.

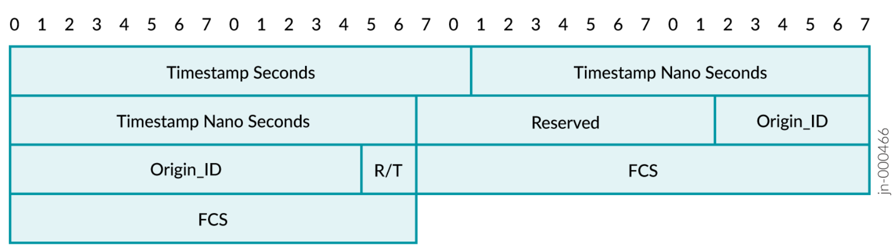
Tailstamps for IFA Probe Packets (QFX5220 only)

The QFX5220 as a transit node can not insert metadata into the metadata stack of the IFA probe packet header. Instead, the QFX5220 adds a tailstamp to the end of the IFA probe packet that includes timestamps and other metadata. The QFX5220 adds a total of 28 bytes of metadata as a tailstamp. Upon receiving the IFA probe packet, the IFA termination node uses the TTL value in the metadata to identify the number of tailstamps (that is, the number of QFX5220 hops on the path between two QFX5120 or QFX5130 devices). Then the tailstamps are converted into the correct metadata format and inserted into the correct place in the metadata stack, so that the metadata appears in the order that the transit nodes added them. Once complete, the IFA termination node exports the data in IPFIX format to the configured external collector.

Due to this inability to insert metadata into the stack, the IFA metadata stack fields `IP TTL` , `Egress Port Speed` and `Congestion` for the QFX5220 are received with the value of 0 at the collector. You must configure the collector to ignore these unsupported fields from the QFX5220.

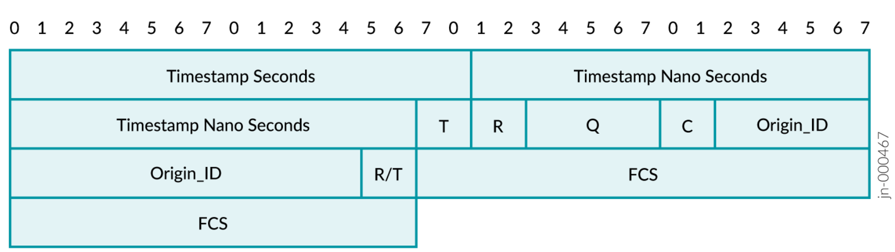
The tailstamp includes 14 bytes of ingress (Rx) tailstamp and 14 bytes of egress (Tx) tailstamp. Figure 48 on page 366 and Figure 49 on page 366 provide details about the format of these timestamps.

Figure 48: Ingress (Rx) Tailstamp Format



Origin_ID[0,9]: Device ID
Origin_ID[10,22]: Port Number

Figure 49: Egress (Tx) Tailstamp Format



Supported Features on IFA Nodes

Table 60 on page 367 lists the features supported by IFA nodes.

Table 60: Supported Features on IFA Nodes

IFA Node	Supported Features
IFA initiator	Traffic and interface types: • IPv4 and IPv6 traffic. • VXLAN traffic. • UDP and TCP. • Tagged and untagged packets. • Aggregation links (LAG) and multichassis LAG (MC-LAG). In case of LAG egress, the original packet and the IFA probe copy use the same port to exit. • IRB interfaces. (The `inband-flow-telemetry-init` and `inband-flow-telemetry-terminate` filters must be applied on the underlying Layer 2 interfaces for IRB traffic.) • ECMP traffic. In case of ECMP traffic, the original packet and the IFA probe copy use the same port to exit. • Interface speeds, such as 10 Gbps, 25 Gbps, 40 Gbps, 50 Gbps, and 100 Gbps.
IFA transit	Identifies IFA packets, appends their metadata, and forwards it.
IFA terminating	• Support to export IFA data to any configured IPv4 collector in IPFIX format. • Support to combine multiple IFA packets into a single IPFIX export.

Supported IFA 2.0 IPFIX Format (Terminating Node)

The terminating node formats the IFA 2.0 packets in IPFIX format, updates the egress port information, and sends the packet to the configured collector. The IFA 2.0 IPFIX template is the same for L3 traffic and VXLAN traffic. Figure 50 on page 368 shows the IPFIX template in which the terminating node formats the IFA 2.0 data and sends it to a collector.

The QFX5240 switch does not use these IPFIX templates. Instead, the QFX5240 switch terminates the IFA packet, mirror encapsulates it, and sends the packet to the collector as a IPFIX PSAMP packet. The PSAMP packet contains the destination MAC address of the collector, the source MAC address of the collector, the source IP address, the destination IP address, the UDP source port, the UDP destination port, the IPFIX PSAMP header, and the MAC source and destination addresses of the IFA packet. These fields are followed by the IP layer header, IFA header, L4 layer (UDP/TCP) header, IFA metadata header and stack, and the payload.

Figure 50: IFA 2.0 IPFIX Template

```
Version: 10
Length: 48
Timestamp: Jan 11, 1970 19:58:15.000000000 IST
FlowSequence: 1128891
Observation Domain Id: 305419896
Set 1 [id=2] (Data Template): 257
  FlowSet Id: Data Template (V10 [IPFIX]) (2)
  FlowSet Length: 32
  Template (Id = 257, Count = 3)
    Template Id: 257
    Field Count: 3
    Field (1/3): 100 [pen: Reserved]
      1... .... .... .... = Pen provided: Yes
      .000 0000 0110 0100 = Type: 100 [pen: Reserved]
      Length: 8
      PEN: Reserved (0)
    Field (2/3): 101 [pen: Reserved]
      1... .... .... .... = Pen provided: Yes
      .000 0000 0110 0101 = Type: 101 [pen: Reserved]
      Length: 65535 [i.e.: "Variable Length"]
      PEN: Reserved (0)
    Field (3/3): 102 [pen: Reserved]
      1... .... .... .... = Pen provided: Yes
      .000 0000 0110 0110 = Type: 102 [pen: Reserved]
      Length: 65535 [i.e.: "Variable Length"]
      PEN: Reserved (0)
```

Figure 51 on page 369 shows a sample VXLAN IFA 2.0 packet received by the configured collector in IPFIX format.

Figure 51: VXLAN IFA 2.0 IPFIX Sample Packet

```
    Version: 10
    Length: 274
  > Timestamp: Jan 11, 1970 19:58:12.000000000 IST
    FlowSequence: 1125782
    Observation Domain Id: 305419896
  ⌄ Set 1 [id=257] (1 flows)
      FlowSet Id: (Data) (257)
      FlowSet Length: 258
      [Template Frame: 330 (received after this frame)]
    ⌄ Flow 1
        Enterprise Private entry: (Reserved) Type 100: Value (hex bytes): 2f 11 00 f0 00 00 f9 10
      ⌄ Enterprise Private entry: (Reserved) Type 101: Value (hex bytes): 10 07 d0 40 0a fc 21 5a 00 01 00 1f 14 d8 f6 80 …
          String_len_short: 255
          String_len_short: 64
      ⌄ Enterprise Private entry: (Reserved) Type 102: Value (hex bytes): 54 4b 8c 1a 05 95 54 4b 8c 19 e7 95 81 00 0f ff …
          String_len_short: 178
```

Limitations of IFA 2.0 Configuration

Before you configure IFA 2.0 on a device running Junos OS, you must be aware of the following limitations:

- **Protocol Number**—IFA 2.0 uses the experimental protocol number 253. If the switch receives any traffic with protocol number 253, those packets will hit the IFA transit filter. In this case the QFX5220 adds a 28-byte tailstamp to those packets. For the QFX5130 and QFX5700 switches, even though the packets hit the filter, IFA metadata is not added to the packets. However, the IFA transit statistics do increment.

- **Filter Resource Allocation**—If filter hardware resources are already exhausted in the system, the IFA feature does not work because it needs filter resources. You can monitor the system log (syslog) for filter space exhaustion errors.

- **Layer 2 and BUM Traffic**—IFA 2.0 is not supported on Layer 2 switched traffic and broadcast, unknown unicast, and multicast (BUM) traffic.

- **IFA Layer 3 and VXLAN Flows**

  - IFA 2.0 supports processing both L3 and VXLAN flows, but you can't configure IFA for both L3 and VXLAN flows on the same device. The `flow-type` options are mutually exclusive. You use the `flow-type` configuration statement to set the flow type of interest —either L3 or VXLAN. This restriction is only applicable for IFA initiator and terminating nodes (generally leaf nodes). For IFA transit nodes (generally spine nodes), it is not required to configure the flow type.

  -

  - For VXLAN IFA flow, the egress port-related metadata for the terminating node (including egress port number, speed, queue ID, and congestion) are incorrect. It is recommended that you ignore the termination node egress-port-related metadata for VXLAN flows.

- For VXLAN flows, IFA terminated packets sent to the collector contain metadata from only the initiator node and all the transit nodes. These packets do not contain metadata from the terminator node.
- An IFA flow-type (L3 or VXLAN) change requires IFA filter removal and reconfiguration. In case of a flow-type mismatch (for example, `flow-type` configured as VXLAN, whereas the incoming traffic is L3 or vice versa), we can't guarantee IFA behavior (IFA probe packets could be initiated with invalid fields).

- IFA Initiator Node
  - L4 header (UDP/TCP) is mandatory for IFA initiation.
  - IFA initiation for VXLAN flow does not work if the egress port is configured to function as a link aggregation group (LAG) (links connecting leaf to spine).
  - You cannot configure different sample rates for different flows on a port for an IFA initiator. All flows within a port must have the same sample rate.
- IFA Transit Nodes—Devices running Junos OS and Junos OS Evolved do not support the maximum length check for the metadata stack. Configure the `hop-limit` option to limit the insertion of metadata on transit nodes. The QFX5220 cannot perform the hop-limit check to insert the tailstamp. The QFX5220 also cannot insert metadata into the metadata stack in the IFA probe packet header; instead, the QFX 5220 appends a tailstamp to the end of the IFA probe packet.

  QFX5220 supports only 18 bits for the `Rx Seconds Timestamp` value. The QFX5130 and QFX5700 support a 20-bit `Rx Seconds Timestamp` value.

  The `Residence Time Nano Seconds` field is updated as the actual value on the QFX5220, QFX5130, and QFX5700 transit nodes, but on the QFX5120 transit node, 1 second (1000000000 ns) is added along with the actual residence time.
- IFA Terminating Node
  - You can configure only a single IPv4 collector at the terminating node.
  - The terminating node metadata has the queue ID 47. This queue ID is reserved for IFA packet export.
  - The terminating node does not perform a hop-limit check. Even if the incoming IFA packet has `hop-limit` set to 0, the terminating node inserts the metadata and reduces the hop limit by 1, which resets the `hop-limit` value to 255.

Usage Considerations

Following are the IFA 2.0 related usage considerations:

- Sampled IFA packets have an additional 40 bytes (4-byte IFA header + 4-byte IFA metadata header + 32-byte metadata) when it egresses on the initiator node. On subsequent IFA nodes, 32-byte IFA metadata is inserted at every hop. Due to insertion of per-hop metadata into IFA packets, the packet size grows after every hop. You must configure the interface's maximum transmission unit (MTU) accordingly along the network path. In case of an IFA zone with a large number of transit nodes, you must take care of the MTU. Alternatively, you can configure the `hop-limit` option at the initiator node to ensure that the size of the IFA packets never exceeds the specified MTU value.

- To select the flow of interest, you can use any combination of source IP address, destination IP address, source port, destination port, and protocol match qualifiers. For VXLAN termination, use the outer source IP address and destination IP address match qualifiers. IFA 2.0 doesn't support any other match qualifiers.

  Another option is to configure the termination filter without any qualifiers. In this case, all incoming IFA packets on the port are terminated.

- You must configure a unique device ID for each hop within an IFA zone. If you've configured the `auto` option for the device ID, then the device ID is generated from the last 20 bits of the router ID or management IP address.

- If you've configured the sampling rate as `aggressive`, the egress ports might experience congestion due to more IFA copies. This port congestion could create congestion on terminating nodes when IFA copies are sent to the chip processor for IPFIX export. We recommend that you select the sampling rate accordingly.

- When you configure an IFA 2.0 initiator, an internal mirror session is created for the loopback port. As a result, the number of user-configurable mirror sessions reduces from 4 to 3.

- The terminating node accepts an IFA packet size up to 9000 bytes (including IFA headers). On the terminating node, multiple IFA received packets are combined into a single IPFIX export packet. You can combine a maximum of 10 IFA records in a single IPFIX export packet. By default, a maximum of 256 bytes of the original flow packet are exported as part of the IPFIX export, along with IFA headers. The maximum size of a single IPFIX packet is 9000 bytes. You must configure the MTU properly on the collector port. Because the maximum size of a single IPFIX packet is 9000 bytes, the maximum clip length for the IPFIX packet is equal to or less than: 9000 bytes - (IFA header length + IFA metadata header length + IFA metadata stack length).

- We recommend that you use only IFA-aware (supported) devices within the IFA zone. We cannot guarantee proper IFA behavior with IFA-unaware devices.

Configure Inband Flow Analyzer 2.0

IN THIS SECTION

IFA is a type of Inband Network Telemetry (INT) that allows you to collect information about the network state by the data plane.

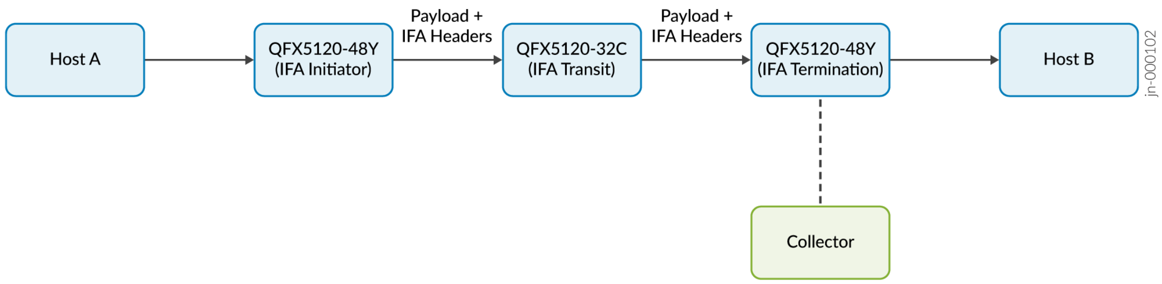
To configure IFA 2.0 for monitoring the network for faults, performance issues, and collect the data for analysis, you need to configure the IFA roles first. You can configure the IFA roles on a Junos OS or Junos OS Evolved device that supports the IFA feature. See Feature Explorer: Inband Flow Analyzer (IFA) 2.0 for supported devices.

Following are some of the guidelines for configuring a Junos OS device for an IFA role:

- You can use the same model switches or different switches to play the IFA roles (initiator, transit, terminator) for a particular IFA flow.
- You can use the same device to perform all three different IFA roles for different flows.
- In an IFA flow, the transit IFA role is optional.

Figure 52 on page 373 illustrates a sample scenario for configuring IFA nodes on Junos OS devices. In this scenario, different Junos OS devices that support the IFA feature play different IFA roles in a single IFA flow.

Figure 52: Sample Inband Flow Analyzer Scenario



Following are some of the guidelines for configuring IFA nodes:

- You can enable the IFA configuration on the interface only through the firewall filter configuration.
- You can apply IFA filter only on ingress direction on the port.

Table 61 on page 373 summarizes the configurations for IFA initiator, transit, and terminating nodes.

Table 61: IFA Configurations for IFA Roles

IFA Configuration Parameter	Configuration Statement	IFA Role
(Mandatory) Configure Device ID	`user@host# set services inband-flow-telemetry device-id (<1 - 1048575> | auto)`	Mandatory configuration for IFA initiator, transit, and terminating nodes.
(Optional, QFX5120-48YM or QFX5220 only) Configure a more accurate clock source	`user@host# set services inband-flow-telemetry clock-source (ntp| ptp)`	IFA initiator, transit, and terminating nodes.
(Optional) IFA maximum metadata stack length	`user@host# set services inband-flow-telemetry meta-data-stack-length <8 - 255>` **Default value** : 240 (for 30 hops)	IFA initiator node

Table 61: IFA Configurations for IFA Roles *(Continued)*

IFA Configuration Parameter	Configuration Statement	IFA Role
(Optional) IFA maximum hop limit	`user@host#` **`set services inband-flow-telemetry hop-limit <1 - 250>`** **Default value** : 250	IFA initiator node
(Optional) No IPv6 address match	`user@host#` **`set services inband-flow-telemetry no-ipv6-address-match`**	IFA initiator/terminating node
(Mandatory) IFA flow type	`user@host#` **`set services inband-flow-telemetry flow-type (l3 | vxlan)`**	Mandatory configuration for IFA initiator and terminating node. This configuration is not required for IFA transit node.
IFA sampling	`user@host#` **`set services inband-flow-telemetry profile`** ***`ifa-profile-name`*** **`sample-rate <1-16777215>`**	IFA initiator node

Table 61: IFA Configurations for IFA Roles *(Continued)*

IFA Configuration Parameter	Configuration Statement	IFA Role
Collector information	`user@host#` **`set services inband-flow-telemetry profile`** ***`ifa-profile-name`*** **`collector source-address`** ***`IP-address`*** `user@host#` **`set services inband-flow-telemetry profile`** ***`ifa-profile-name`*** **`collector destination-address`** ***`IP-address`*** `user@host#` **`set services inband-flow-telemetry profile`** ***`ifa-profile-name`*** **`collector destination-port`** ***`port-number`*** `user@host#` **`set services inband-flow-telemetry profile`** ***`ifa-profile-name`*** **`collector maximum-clip-length`** ***`length`*** `user@host#` **`set services inband-flow-telemetry profile`** ***`ifa-profile-name`*** **`collector mtu`** ***`size`***	IFA terminating node
IFA filter for L3 flow	For example: `user@host#` **`set firewall family inet filter f1 term t1 from`** ***`match-condition`*** `user@host#` **`set firewall family inet filter f1 term t1 then inband-flow-telemetry-init p1`** `user@host#` **`set firewall family inet filter f1 term t2 from`** ***`match-condition`*** `user@host#` **`set firewall family inet filter f1 term t2 then inband-flow-telemetry-terminate p2`** `user@host#` **`set interfaces (`*`interface-name`* `|` *`wildcard`*`) unit 0 family inet filter input f1`**	IFA initiator/terminating node

Table 61: IFA Configurations for IFA Roles *(Continued)*

IFA Configuration Parameter	Configuration Statement	IFA Role
IFA filter for VXLAN flow	For example: user@host# **set firewall family ethernet-switching filter f1 term term1 from** ***match-condition*** user@host# **set firewall family ethernet-switching filter f1 term t1 then inband-flow-telemetry-init p1** user@host# **set firewall family ethernet-switching filter f1 term t2 from** ***match-condition*** user@host# **set firewall family ethernet-switching filter f1 term t2 then inband-flow-telemetry-terminate p2** user@host# **set interfaces (*interface-name* | *wildcard*) unit 0 family ethernet-switching filter input f1**	IFA initiator/terminating node

Configure IFA Initiator Node

To configure your device as IFA 2.0 initiator:

1. Configure the device ID. You can also configure the value `auto` for `device-id`. If the `device-id` is configured as `auto`, the `device-id` is internally generated from the router ID or the management IP address.

   ```
   user@host# set services inband-flow-telemetry device-id (id-number | auto)
   ```

   In this example, the device id for IFA initiator node is configured as 10000.

   ```
   user@host# set services inband-flow-telemetry  device-id 10000
   ```

2. Configure the flow type. You can configure either of two flow types, `l3` or `vxlan`. You cannot configure L3 and VXLAN flows together in the same device.

```
user@host# set services inband-flow-telemetry flow-type (l3 | vxlan)
```

In this example, the flow type is configured as `l3`. If you configure `l3` flow-type in the initiator node, then you must choose `l3` flow-type for the terminating node also.

```
user@host# set services inband-flow-telemetry flow-type l3
```

3. (Optional) Configure the maximum metadata stack length. Each IFA hop inserts hop-specific metadata into the IFA metadata stack.

```
user@host# set services inband-flow-telemetry meta-data-stack-length value
```

In this example, the metadata stack length is configured as 80.

```
user@host# set services inband-flow-telemetry meta-data-stack-length 80
```

4. Configure the hop limit.

```
user@host# set services inband-flow-telemetry hop-limit value
```

In this example, `hop-limit` is configured as 10. The hop limit is decremented at each hop. If the incoming hop limit is 0, the current node does not insert metadata.

```
user@host# set services inband-flow-telemetry hop-limit 10
```

5. Configure IFA sampling. The sampling rate is the average number of samples obtained in one second. You cannot have different sample rate for different flows on an IFA initiator node enabled on a port. All flows within a port should have same sample rate.

```
user@host# set services inband-flow-telemetry profile ifa-profile-name sample-rate value
```

In this example, the sample rate is configured as 1000; meaning out of 1000 packets, 1 packet will be sampled per second.

```
user@host# set services inband-flow-telemetry profile p1 sample-rate 1000
```

6. Configure IFA firewall filters. You can configure firewall filter with any of the below match conditions:

   - Source IP address
   - Destination IP address
   - Source port
   - Destination port
   - Protocol

   Create a firewall and configure the action `inband-flow-telemetry-init`.

   ```
   user@host# set firewall family inet filter filter-name term term-name from source-address
   ipv4-address
   user@host# set firewall family inet filter filter-name term term-name from destination-
   address ipv4-address
   user@host# set firewall family inet filter filter-name term term-name then inband-flow-
   telemetry-init ifa-profile-name
   ```

   In this example, you configure a firewall filter named `f1`, with the term name `t1` containing the action `inband-flow-telemetry-init`, and the inband flow telemetry initiator profile `p1` mapped to it:

   ```
   user@host# set firewall family inet filter f1 term t1 from source-address 10.30.1.4/32
   user@host# set firewall family inet filter f1 term t1 from destination-address 10.40.1.4/32
   user@host# set firewall family inet filter f1 term t1 then inband-flow-telemetry-init p1
   ```

7. Map the firewall filter to the family under the logical unit of the already-configured interface to apply the action `inband-flow-telemetry-init` in the ingress direction.

   To map the firewall filter:

   ```
   user@host# set interfaces interface-name unit 0 family inet filter input filter-name
   ```

In this example, you map the `f1` firewall filter to the `inet` family of logical interface 0 of the physical interface et-0/0/0:

```
user@host# set interfaces et-0/0/0 unit 0 family inet filter input f1
```

Configure IFA Transit Node

To configure your device as IFA transit node:

Configure the device ID. You can also configure the value `auto` for `device-id`. If the `device-id` is configured as `auto`, then the `device-id` is internally generated from the router ID or the management IP address.

```
user@host# set services inband-flow-telemetry device-id (id-number | auto)
```

For example:

```
user@host# set services inband-flow-telemetry device-id 10001
```

Configure IFA Terminating Node

To configure your device as IFA terminating node:

1. Configure the device ID. You can also configure the value `auto` for `device-id`. If the `device-id` is configured as `auto`, then the `device-id` is internally generated from the router ID or the management IP address.

   ```
   user@host# set services inband-flow-telemetry device-id (id-number | auto)
   ```

   For example:

   ```
   user@host# set services inband-flow-telemetry device-id 10002
   ```

2. Configure the flow type. You can configure either of two flow types, `l3` or `vxlan`. You cannot configure L3 and VXLAN flows together in the same device.

   ```
   user@host# set services inband-flow-telemetry flow-type (l3 | vxlan)
   ```

If you configure `l3` flow-type in the initiator node, then you must choose `l3` flow-type for the terminating node also.

```
user@host# set services inband-flow-telemetry flow-type l3
```

3. Configure IFA profile with the collector information for the terminating node.

```
user@host#
user@host# set services inband-flow-telemetry profile ifa-profile-name collector source-
address ipv4-address
user@host# set services inband-flow-telemetry profile ifa-profile-name collector destination-
address ipv4-address
user@host# set services inband-flow-telemetry profile ifa-profile-name collector destination-
port port-number
```

For example:

```
user@host# set services inband-flow-telemetry profile p2 collector source-address 10.50.1.1
user@host# set services inband-flow-telemetry profile p2 collector destination-address
10.60.1.1
user@host# set services inband-flow-telemetry profile p2 collector destination-port 2055
```

4. You can configure firewall filter with any of the below match conditions:

- Source IP address
- Destination IP address
- Source port
- Destination port
- Protocol

Create a firewall and configure the action `inband-flow-telemetry-terminate`.

```
user@host# set firewall family inet filter filter-name term term-name from source-address
ipv4-address
user@host# set firewall family inet filter filter-name term term-name from destination-
address ipv4-address
user@host# set firewall family inet filter filter-name term term-name then inband-flow-
telemetry-terminate ifa-profile-name
```

```
user@host# set firewall interfaces interfaces-name unit logical-unit-number family inet
filter input filter-name
```

In this example, you configure a firewall filter named `f2`, with the term name `t1` containing the action `inband-flow-telemetry-terminate`, and the inband-flow-telemetry-terminate profile `p2` mapped to it:

```
user@host# set firewall family inet filter f2 term t1 from source-address 10.30.1.4/32
user@host# set firewall family inet filter f2 term t1 from destination-address 10.40.1.4/32
user@host# set firewall family inet filter f2 term t1 then inband-flow-telemetry-terminate p2
```

5. Map the firewall filter to the family under the logical unit of the already-configured interface to apply the `inband-flow-telemetry-terminate` action in the egress direction.

   To map the firewall filter:

```
user@host# set interfaces interface-name unit 0 family inet filter input filter-name
```

   In this example, you map the `f2` firewall filter to the `inet` family of the logical interface 0 of the physical interface et-0/0/0:

```
user@host# set interfaces et-0/0/0 unit 0 family inet filter input f2
```

View Inband Flow Analyzer Statistics

You can view the following IFA related information:

- IFA statistics using the `show services inband-flow-telemetry stats` operational mode command.
- IFA global parameters using the `show services inband-flow-telemetry global` operational mode command.
- IFA-configured profiles using the `show services inband-flow-telemetry profile` operational mode command.

You can clear the IFA statistics using `clear inband-flow-telemetry stats` operational mode command.

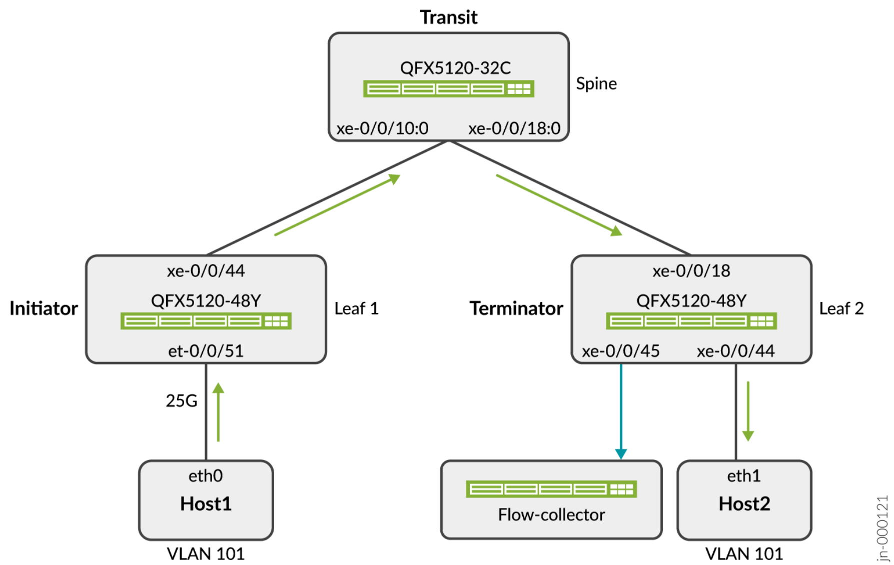
IFA statistics are retrieved directly from the PFE and are not maintained in the Routing Engine. Therefore, a PFE-process restart clears the IFA statistics and a Routing-Engine process restart does not impact the IFA statistics.

Example - Configure Inband Flow Analyzer 2.0 for Traffic Monitoring

IN THIS SECTION

Use this example to configure the IFA 2.0 nodes on your QFX Series switches that enable analyzing of Layer 3 or VXLAN traffic flows. Figure 53 on page 383 shows the topology where IFA 2.0 is configured on QFX Series switches that support the IFA 2.0 feature. In this topology, VXLAN traffic is monitored at the initiator and data is collected at the terminating node for analysis.

Figure 53: Topology for Analyzing VXLAN Traffic Flow using IFA 2.0



Requirements

This example uses the following hardware and software components:

- One QFX5120-32C switch as a spine node
- Two QFX5120-48Y switches as the leaf nodes
- Junos OS Release 21.4R1

Pre-Requisites

This example assumes that you already have an EVPN-VXLAN based network and want to enable traffic monitoring on QFX switches.

Before you Begin

- Make sure you understand how EVPN and VXLAN works. See Example: Configuring IRB Interfaces in an EVPN-VXLAN Environment to Provide Layer 3 Connectivity for Hosts in a Data Center and Bridged Overlay Design and Implementation to understand EVPN-VXLAN in detail.
- For IFA terminating node configurations to take effect you need to have a valid Advanced Telemetry Feature (ATF) license in place.

Overview

In this example, you'll configure one of the QFX5120-48Y switches (Leaf 1) as an initiator node, the QFX5120-32C switch as a transit node, and the second QFX5120-48Y switch (Leaf 2) as a terminating node. The VXLAN traffic flows from Host 1 to Host 2. Configuring IFA on the ingress and egress nodes allows you to monitor network operation and identify the performance issues.

The QFX5120-32C functions as a spine to connect the QFX5120-48Y leaf nodes. At the terminating node, you collect the sampled traffic in IPFIX format using an IPv4 collector application.

Configuration

In this example, you'll configure the following functionality on the switches:

1. Configure Leaf 1 as an initiator node and configure initiator related attributes, like global device identifier and the sampling rate. Configure an IFA profile and firewall filter with the action as `inband-flow-telemetry-init`, and bind the IFA firewall filter to the interfaces.

2. Configure the QFX5120-32C spine switch as a transit node with a global device identifier. When you configure a global device identifier, the spine device adds the IFA metadata and forwards the IFA probe packets.

3. Configure Leaf 2 as a terminating node. Configure the IFA profile with the collector information and firewall filter with the action as `inband-flow-telemetry-terminate`, and bind the IFA firewall filter to the interfaces.

CLI Quick Configuration

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

Configuration on QFX5120-48Y Switch (Leaf 1 — IFA Initiator Node)

> **NOTE**: Recall that in this example you add IFA to a pre-configured EVPN-VXLAN baseline. The configuration shown here focuses on the delta needed to add IFA to the baseline. We show some of the existing configuration to best show how the IFA delta relates to the baseline.

```
set services inband-flow-telemetry device-id 15000
set services inband-flow-telemetry meta-data-stack-length 100
set services inband-flow-telemetry hop-limit 4
set services inband-flow-telemetry flow-type vxlan
```

```
set services inband-flow-telemetry profile ifa_profile_host1 sample-rate 1

set interfaces et-0/0/51:0 unit 0 family ethernet-switching filter input f_init

set firewall family ethernet-switching filter f_init term t1 from ip-protocol udp
set firewall family ethernet-switching filter f_init term t1 from ip-protocol tcp
set firewall family ethernet-switching filter f_init term t1 then inband-flow-telemetry-init
ifa_profile_host1
set firewall family ethernet-switching filter f_init term t1 then count ifa_stats
set firewall family ethernet-switching filter f_init term t1 then accept
set firewall family ethernet-switching filter f_init term t2 then count non_ifa_stats
set firewall family ethernet-switching filter f_init term t2 then accept
```

Configuration on QFX5120-32C Switch (IFA Transit Node)

```
set services inband-flow-telemetry device-id 15001
```

Configuration on QFX5120-48Y Switch (Leaf 2 — IFA Terminating Node)

```
set services inband-flow-telemetry device-id 15002
set services inband-flow-telemetry meta-data-stack-length 100
set services inband-flow-telemetry hop-limit 5
set services inband-flow-telemetry flow-type vxlan
set services inband-flow-telemetry profile p_term collector source-address 172.16.3.1
set services inband-flow-telemetry profile p_term collector destination-address 172.16.3.2
set services inband-flow-telemetry profile p_term collector destination-port 3055

set interfaces xe-0/0/18 unit 0 family inet filter input f_term

set interfaces xe-0/0/45 description To_Collector
set interfaces xe-0/0/45 unit 0 family inet address 172.16.3.1/24

set firewall family inet filter f_term term ifa then inband-flow-telemetry-terminate p_term
set firewall family inet filter f_term term ifa then count ifa_term
set firewall family inet filter f_term term other then count non_ifa_term
set firewall family inet filter f_term term other then accept
```

Step-by-Step Procedure

Configure QFX5120-48Y Switch (Leaf 1) as an Initiator Node

An IFA initiator node performs the following functions for a flow:

- Samples the flow traffic of interest based on the configuration.
- Converts the traffic into an IFA flow by adding an IFA header to each sample.
- Updates the packet with initiator node metadata.

1. Configure the IFA initiator node attributes. The traffic flow type is configured as VXLAN for initiator node. Note that you must configure the same flow type for both the initiator and the terminating node, either L3 or VXLAN. As in this example, if the VXLAN traffic flow type is configured for the initiator node, ensure that you configure VXLAN traffic flow type for the terminating node as well.

```
[edit]
user@host# set services inband-flow-telemetry device-id 15000
user@host# set services inband-flow-telemetry meta-data-stack-length 100
user@host# set services inband-flow-telemetry hop-limit 4
user@host# set services inband-flow-telemetry flow-type vxlan
user@host# set services inband-flow-telemetry profile ifa_profile_host1 sample-rate 1
```

   When `sample-rate` is configured with value as 1, every packet that is received in the ingress port is sampled. If you prefer less aggressive sampling, increase the `sample-rate` value.

2. Bind the filter to the initiator node ingress interface.

```
[edit]
user@host# set interfaces et-0/0/51:0 unit 0 family ethernet-switching filter input f_init
```

3. Create a firewall to control IFA sampling. You begin by defining the types of host traffic that should be sampled. In this example you want to perform analysis on UDP and TCP traffic flows. In this example, you configure an firewall filter named `f_init`, with the term name `term1`.

```
[edit]
user@host# set firewall family ethernet-switching filter f_init term t1 from ip-protocol udp
user@host# set firewall family ethernet-switching filter f_init term t1 from ip-protocol tcp
user@host# set firewall family ethernet-switching filter f_init term t1 then accept
```

You configure the filter to perform IFA sampling by adding the action modifier `inband-flow-telemetry-init` to the *t1* term. Note that the inband flow telemetry profile `ifa_profile_host1` is linked to the filter:

```
user@host# set firewall family ethernet-switching filter f_init term t1 then inband-flow-
telemetry-init ifa_profile_host1
user@host# set firewall family ethernet-switching filter f_init term t1 then count ifa_stats
user@host# set firewall family ethernet-switching filter f_init term t2 then count
non_ifa_stats
user@host# set firewall family ethernet-switching filter f_init term t2 then accept
```

Configure QFX5120-32C Switch as a Transit Node

An IFA transit node inserts transit node metadata in the IFA packets in the specified VXLAN flow.

Configure the global device identifier for the transit node, QFX5120-32C switch.

```
user@host# set services inband-flow-telemetry device-id 15001
```

Configure QFX5120-48Y Switch (Leaf 2) as a Terminating Node

An IFA terminating node performs the following for a flow:

- Inserts terminating node metadata in IFA packets.
- Performs a local analytics function on one or more segments of metadata, for example, threshold breach for residence time, congestion notifications, and so on.
- Filters an IFA flow in case of cloned traffic.
- Sends a copy or report of the packet to collector.
- Removes the IFA headers and forwards the packet in case of live traffic.

1. Configure the terminating node related attributes, like global device identifier and flow type.

```
user@host# set services inband-flow-telemetry device-id 15002
user@host# set services inband-flow-telemetry meta-data-stack-length 100
user@host# set services inband-flow-telemetry hop-limit 5
user@host# set services inband-flow-telemetry flow-type vxlan
```

Configure an IFA profile with the collector related information.

```
user@host# set services inband-flow-telemetry profile p_term collector source-address
172.16.3.1
user@host# set services inband-flow-telemetry profile p_term collector destination-address
172.16.3.2
user@host# set services inband-flow-telemetry profile p_term collector destination-port 3055
```

2. Configure the collector interface for terminating node Leaf 2.

```
user@host# set interfaces xe-0/0/45 unit 0 family inet address 172.16.3.1/24
```

Apply the firewall filter to the pre-configured interface to activate inband flow telemetry egress processing at Leaf 2.

In this example, you map the `f-term` firewall filter to the `inet` family of logical interface 0 of the physical interface xe-0/0/18:

```
user@host# set interfaces xe-0/0/18 unit 0 family inet filter input f_term
```

3. Create a firewall filter and configure the action `inband-flow-telemetry-terminate`.

In this example, you configure a firewall filter named `f-term`, with the term name `t1` containing the action `inband-flow-telemetry-terminate`, with the inband flow telemetry terminate profile `p_term` mapped to it:

```
user@host# set firewall family inet filter f_term term t1 then count ifa_term
user@host# set firewall family inet filter f_term term t1 then inband-flow-telemetry-
terminate p_term
user@host# set firewall family inet filter f_term term t1 then accept
user@host# set firewall family inet filter f_term term other then count non_ifa_term
user@host# set firewall family inet filter f_term term other then accept
```

Results

Results on QFX5120-48Y Switch (Leaf 1 — IFA Initiator Node)

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

> **NOTE**: The output shows portions of the pre-existing EVPN-VXLAN baseline to provide the context for the configuration delta needed to add IFA.

```
[edit]
user@host> show configuration services
inband-flow-telemetry {
    device-id {
        15000;
    }
    meta-data-stack-length 100;
    hop-limit 4;
    flow-type vxlan;
    profile {
        ifa_profile_host1 {
            sample-rate 1;
        }
    }
}
```

```
[edit]
user@host> show configuration interfaces
[output truncated]
xe-0/0/44 {
    description Connected_to_Spine1;
    unit 0 {
        family inet {
            address 10.100.13.1/24;
        }
    }
}
et-0/0/51:0 {
    description Connected_to_Host1_vlan_101;
```

```
    unit 0 {
        family ethernet-switching {
            interface-mode trunk;
            vlan {
                members 101;
            }
            filter {
                input f_init;
            }
        }
    }
}
[output truncated]
```

```
[edit]
user@host> show configuration firewall
family ethernet-switching {
    filter f_init {
        term t1 {
            from {
                ip-protocol [ udp tcp ];
            }
            then {
                accept;
                inband-flow-telemetry-init ifa_profile_host1;
                count ifa_stats;
            }
        }
        term t2 {
            then {
                accept;
                count non_ifa_stats;
            }
        }
    }
}
```

When you are done configuring the feature on your device, enter `commit` from configuration mode.

Results on QFX5120-32C Switch (IFA Transit Node)

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

```
[edit]
user@host> show configuration services
inband-flow-telemetry {
    device-id {
        15001;
    }
}
```

When you are done configuring the feature on your device, enter `commit` from configuration mode.

Results on QFX5120-48Y Switch (Leaf 1 — IFA Terminating Node)

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

```
[edit]
user@host> show configuration services
inband-flow-telemetry {
    device-id {
        15002;
    }
    meta-data-stack-length 100;
    hop-limit 5;
    flow-type vxlan;
    profile {
        p_term {
            collector {
                source-address 172.16.3.1;
                destination-address 172.16.3.2;
                destination-port 3055;
            }
        }
    }
}
```

[edit]

user@host> **show configuration interfaces**

```
[edit]
user@host> show configuration interfaces
[output truncated]
xe-0/0/18 {
    description Connected_to_Spine1;
    unit 0 {
        family inet {
            filter {
                input f_term;
            }
            address 10.100.12.1/24;
        }
    }
}
xe-0/0/44 {
    description Connected_to_Host2_vlan_101;
    unit 0 {
        family ethernet-switching {
            interface-mode trunk;
            vlan {
                members 101;
            }
        }
    }
}
xe-0/0/45 {
    description To_Collector;
    mtu 9200;
    unit 0 {
        family inet {
            address 172.16.3.1/24;
        }
    }
}
[output truncated]
```

```
[edit]
user@host> show configuration firewall
family inet {
```

```
    filter f_term {
        term t1 {
            then {
                count ifa_term_c;
                inband-flow-telemetry-terminate p_term;
                accept;
            }
        }
        term other {
            then {
                count non_ifa_term;
                accept;
            }
        }
    }
}
```

When you are done configuring the feature on your device, enter `commit` from configuration mode.

Verification

Verification on QFX5120-48Y Switch (Leaf 1 — IFA Initiator Node)

Verify IFA Statistics

Purpose

Display the IFA statistics on the initiator node.

Action

From operational mode, enter the `show services inband-flow-telemetry stats` command.

```
IFA Init Packets            : 70989449712
IFA Transit Packets         : 0
IFA Terminate Rx Packets    : 0
IFA Terminate Tx Packets    : 0
```

Verify IFA Global Configuration

Purpose

Display the IFA global parameters configured on the initiator node.

Action

From operational mode, enter the `show services inband-flow-telemetry global` command.

```
Global Device ID              : 15000
Meta-data Stack Length        : 100
Hop Limit                     : 4
Flow Type                     : vxlan
```

Verify IFA Profile

Purpose

Display the IFA profile configured on the initiator node.

Action

From operational mode, enter the `show services inband-flow-telemetry profile` command.

```
Profile Name              : ifa_profile_host1
Sample rate               : 1
Source Address            : 0.0.0.0
Destination Address       : 0.0.0.0
Destination Port          : 0
```

Verification on QFX5120-32C Switch (IFA Transit Node)

Verify IFA Statistics

Purpose

Display the IFA statistics on the transit node.

Action

From operational mode, enter the `show services inband-flow-telemetry stats` command.

```
IFA Init Packets              : 0
IFA Transit Packets           : 26057387140
IFA Terminate Rx Packets      : 0
IFA Terminate Tx Packets      : 0
```

Verify IFA Global Configuration

Purpose

Display the IFA global parameters configured on the transit node.

Action

From operational mode, enter the `show services inband-flow-telemetry global` command.

```
Global Device ID              : 15001
Meta-data Stack Length        : 240
Hop Limit                     : 250
Flow Type                     : NA
```

Verification on QFX5120-48Y Switch (Leaf 2 — IFA Terminating Node)

Verify IFA Statistics

Purpose

Display the IFA statistics on the terminating node.

Action

From operational mode, enter the `show services inband-flow-telemetry stats` command.

```
IFA Init Packets              : 0
IFA Transit Packets           : 373569
IFA Terminate Rx Packets      : 374448690
IFA Terminate Tx Packets      : 41605188
```

Verify IFA Global Configuration

Purpose

Display the IFA global parameters configured on the terminating node.

Action

From operational mode, enter the `show services inband-flow-telemetry global` command.

```
Global Device ID              : 15002
Meta-data Stack Length        : 100
Hop Limit                     : 5
Flow Type                     : vxlan
```

Verify IFA Profile

Purpose

Display the IFA profile configured on the terminating node.

Action

From operational mode, enter the `show services inband-flow-telemetry profile` command.

```
Profile Name              : p_term
Sample rate               : 0
Source Address            : 172.16.3.1
Destination Address       : 172.16.3.2
Destination Port          : 3055
```

SEE ALSO

inband-flow-telemetry

clear inband-flow-telemetry stats

show services inband-flow-telemetry

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
25.4R1-EVO	We've added support for configuring live mode to Junos OS Evolved, to the QFX5240 switch.
24.4R1-EVO	We've extended support for the initiator and terminator roles to Junos OS Evolved.
22.4R1-EVO	We've extended support for the IFA 2.0 transit node role to Junos OS Evolved.
22.2R1	We've added support for configuring the MTU and maximum clip length for IFA packets, and for the QFX5120-48YM switch, setting the IFA clock source.
21.4R1	We've introduced support for IFA 2.0 on QFX Series switches. IFA 2.0 monitors and analyzes packets when they enter and exit the network. You can use IFA 2.0 to monitor the network for faults and performance bottlenecks. IFA 2.0 supports both Layer 3 and VXLAN flows.

CHAPTER 11

Juniper Resiliency Interface

IN THIS CHAPTER

Juniper Resiliency Interface

SUMMARY

For MX Series routers with MPC or LC480 line cards and for PTX Series routers running Junos OS Evolved, you can configure the Juniper Resiliency Interface (JRI) to detect, correlate, and mitigate exceptions and thereby reduce the mean time to repair (MTTR) for issues. For forwarding exceptions, JRI also extends the inline monitoring services feature with Juniper-specific IPFIX information elements (IEs) for exception data and introduces the concept of an observation cloud, which is a set of observation domains. You can send the IPFIX packets to either an on-box or an off-box collector.

IN THIS SECTION

Understand Juniper Resiliency Interface

Packets that need to be forwarded to the adjacent network element or a neighboring device along a routing path might be dropped by a router owing to several factors. Every network encounters issues, such as packet loss, from time to time. Some of the causes for such a loss of traffic or a block in transmission of data packets include: overloaded system conditions, profiles and policies that restrict the bandwidth or priority of traffic, network outages, or disruption by physical cable faults. Packet loss also happens because of incorrect stitching of the forwarding path or a mismatch between the control plane state and the data plane state. You could use counters and metrics from `show` commands to diagnose and debug network performance, but doing so can be tedious and time-consuming. JRI reports exception data from entities in the system which encounter packet drops, enabling you to automate the workflow involved in detecting, reporting and mitigating adverse exceptions.

For operating system and routing exceptions, the exception data is reported in telemetry key-value pairs.

For forwarding exceptions, the exception data is reported in IPFIX packets. The IEs in the IPFIX primary data record packet capture the following data:

- Exception reason (for example, firewall discard)
- Packet direction (ingress or egress)
- First N bytes of the packet
- Ingress interface
- Egress interface
- Next-hop identifier (Junos OS only)

Table 62 on page 398 shows the format of the IPFIX Primary Data Record with the Juniper-specific IEs.

Table 62: IPFIX Primary Data Record

IE Name	IE Identifier	Description	Length (in Bytes)
forwardingClassandDropPriority	Observation Cloud Common Property ID (CPID)—IE 137, a set of common properties that is locally unique per Observation Cloud	Forwarding class and drop priority ID	4
forwardingExceptionCode	Observation Cloud CPID—IE 137	Exception code that causes packet drops OR is zero when the exception is not met or set; a 16-bit unsigned integer number	2
forwardingNextHopId	Observation Cloud CPID—IE 137	(Junos OS only) Unicast next-hop Index used for forwarding	4

Table 62: IPFIX Primary Data Record *(Continued)*

IE Name	IE Identifier	Description	Length (in Bytes)
egressInterfaceIndex	Observation Cloud CPID—IE 137	Index of egress logical interface when flowDirection=output, otherwise 0	4
underlyingIngressInterfaceIndex	Observation Cloud CPID—IE 137	(Junos OS only) Index of underlying layer 2 ingress logical interface, wherever applicable (for example, AE and IRB cases—see *primary-data-record-fields* for more information)	4
ingressInterfaceIndex	Observation Cloud CPID—IE 137	Index of ingress logical interface	4
ingressInterface	IE 10	SNMP index of ingress logical interface	4
egressInterface	IE 14	SNMP index of egress logical interface when flowDirection=output, otherwise 0	4
flowDirection	IE 61	Direction (0: input, 1:output)	1
dataLinkFrameSize	IE 312	Length of sampled data link frame	2
dataLinkFrameSection	IE 315	N octets from the data link frame of the monitored packet	variable

Limitations:

- Exceptions are collected and exported on a best-effort basis.
- Any limitations or caveats for inline monitoring services also apply to JRI, because JRI uses inline monitoring services to sample and collect the packets.

- All dropped packets cannot be sampled and profiled. Classes of exceptions are sampled at the default sampling rate, unless you configure this rate with the `sampling-rate` statement at either the `[edit services inline-monitoring instance` *`instance-name`* `collector` *`collector-name`*`]` hierarchy level (Junos OS) or at the `[edit services inline-monitoring instance` *`instance-name`*`]` hierarchy level (Junos OS Evolved). Junos OS allows the sampling rate to be configured per collector, allowing different rates for each collector; Junos OS Evolved allows one sampling rate per inline-monitoring instance.
- For exception reporting in the egress direction, the layer 2 header or any encapsulation header is not included in IE-315, dataLinkFrameSelection, because exceptions happen before layer 2 or tunnel encapsulation.
- For exception reporting in the egress direction, the receiver of the IPFIX packet must ignore IE-312, dataLinkFrameSize, because the field does not have the correct value.
- For the egress direction, you cannot configure both sFlow and exception reporting on the same interface.
- Inline-monitoring instance actions and firewall re-direct instance actions are not supported in the same term of the firewall filter. (Junos OS Evolved)
- Inline-monitoring instance actions and port-mirroring instance actions are not supported in the same term of the firewall filter. (Junos OS Evolved)
- For collectors, you cannot configure routing instances, DSCP bits, or forwarding class. (Junos OS Evolved)
- For more information about the Juniper-specific IEs, including caveats and limitations, see *primary-data-record-fields*.

Configure JRI for Operating System and Routing Exceptions

To configure JRI for operating system and routing exceptions:

1. Subscribe to the Junos Telemetry Interface XPaths:

   Notifications are exported using gRPC/gNMI to an off-box collector.

   For Junos OS:

   ```
   /junos/exception-profiles/routing-profile
   /junos/exception-profiles/os-profile/
   ```

   For Junos OS Evolved (routing exceptions only):

   ```
   /junos/exception-profiles/routing-profile
   ```

2. (Optional) Additionally, if you prefer to use the on-box collector instead of sending the data to an off-box collector, then configure an on-box storage location for the exception data.

   To configure:

   ```
   user@host# set system resiliency exceptions exception-type
   user@host# set system resiliency store file file-name
   user@host# set system resiliency store file size file-size
   ```

   In this example, you configure the file in which to store the exception data:

   For Junos OS:

   ```
   user@host# set system resiliency exceptions routing
   user@host# set system resiliency exceptions os
   user@host# set system resiliency store file file1
   user@host# set system resiliency store size 1g
   ```

   For Junos OS Evolved (routing exceptions only):

   ```
   user@host# set system resiliency exceptions routing
   user@host# set system resiliency store file file1
   user@host# set system resiliency store size 1g
   ```

Configure JRI for Forwarding Exceptions

To configure JRI for forwarding exceptions:

1. Define the IPFIX template.

   To configure attributes of the template:

   For Junos OS:

   ```
   user@host# set services inline-monitoring template template_1 template-refresh-rate template-
   refresh-rate
   user@host# set services inline-monitoring template template_1 template-id template-identifier
   user@host# set services inline-monitoring template template_1 primary-data-record-fields
   primary-data-record-field-name
   ```

In this example, the template refresh rate is set to 30 seconds, you've configured a template identifier, and you've configured the fields of the primary data record:

```
user@host# set services inline-monitoring template template_1 template-refresh-rate 30
user@host# set services inline-monitoring template template_1 template-id 1024
user@host# set services inline-monitoring template template_1 primary-data-record-fields cpid-
ingress-interface-index
user@host# set services inline-monitoring template template_1 primary-data-record-fields cpid-
underlying-interface-index
user@host# set services inline-monitoring template template_1 primary-data-record-fields cpid-
egress-interface-index
user@host# set services inline-monitoring template template_1 primary-data-record-fields cpid-
forwarding-nexthop-id
user@host# set services inline-monitoring template template_1 primary-data-record-fields cpid-
forwarding-exception-code
user@host# set services inline-monitoring template template_1 primary-data-record-fields cpid-
forwarding-class-drop-priority
user@host# set services inline-monitoring template template_1 primary-data-record-fields
ingress-interface-snmp-id
user@host# set services inline-monitoring template template_1 primary-data-record-fields
egress-interface-snmp-id
user@host# set services inline-monitoring template template_1 primary-data-record-fields
direction
```

For Junos OS Evolved, the system generates the template ID and the software supports most of the fields of the primary data record:

```
user@host# set services inline-monitoring template template_1 template-refresh-rate template-
refresh-rate
user@host# set services inline-monitoring template template_1 primary-data-record-fields
primary-data-record-field-name
```

In this example, the template refresh rate is set to 30 seconds and you've configured the fields of the primary data record:

```
user@host# set services inline-monitoring template template_1 template-refresh-rate 30
user@host# set services inline-monitoring template template_1 template-id 1024
user@host# set services inline-monitoring template template_1 primary-data-record-fields cpid-
ingress-interface-index
user@host# set services inline-monitoring template template_1 primary-data-record-fields cpid-
egress-interface-index
```

```
user@host# set services inline-monitoring template template_1 primary-data-record-fields cpid-
forwarding-exception-code
user@host# set services inline-monitoring template template_1 primary-data-record-fields cpid-
forwarding-class-drop-priority
user@host# set services inline-monitoring template template_1 primary-data-record-fields
ingress-interface-snmp-id
user@host# set services inline-monitoring template template_1 primary-data-record-fields
egress-interface-snmp-id
user@host# set services inline-monitoring template template_1 primary-data-record-fields
direction
```

2. Attach the template to the instance and describe the collector.

   Junos OS and Junos OS Evolved differ in how to achieve this step. To configure the instance and collector:

   For Junos OS:

```
user@host# set services inline-monitoring instance instance-name template-name template-name
user@host# set services inline-monitoring instance instance-name collector collector-name
source-address IPv4-address
user@host# set services inline-monitoring instance instance-name collector collector-name
destination-address IPv4-address
user@host# set services inline-monitoring instance instance-name collector collector-name
dscp dscp-bits
user@host# set services inline-monitoring instance instance-name collector collector-name
destination-port port
```

   In this example, you create a template with the name `template_1`, create an inline-monitoring instance `i1`, and create the configuration for the on-box collector `c2`. For an on-box collector for Junos OS, the destination address must be a local address and the destination port must be port 4739. For an off-box collector for Junos OS, specify the destination address and port for that collector.

   For Junos OS:

```
user@host# set services inline-monitoring instance i1 template-name template_1
user@host# set services inline-monitoring instance i1 collector c2 source-address 10.11.12.1
user@host# set services inline-monitoring instance i1 collector c2 destination-address
10.11.12.2
user@host# set services inline-monitoring instance i1 collector c2 dscp 21
user@host# set services inline-monitoring instance i1 collector c2 destination-port 4739
```

For Junos OS Evolved, you cannot configure the DSCP bits, but the process is otherwise the same as in Junos OS for an off-box collector:

```
user@host# set services inline-monitoring instance instance-name template-name template-name
user@host# set services inline-monitoring instance instance-name collector collector-name
source-address IPv4-address
user@host# set services inline-monitoring instance instance-name collector collector-name
destination-address IPv4-address
user@host# set services inline-monitoring instance instance-name collector collector-name
destination-port port
```

For Junos OS Evolved, for an on-box collector, you configure the `controller re` statement instead of a local destination address and port, and you cannot configure the DSCP bits:

```
user@host# set services inline-monitoring instance instance-name template-name template-name
user@host# set services inline-monitoring instance instance-name collector collector-name
source-address IPv4-address
user@host# set services inline-monitoring instance instance-name controller re
```

In this example, for Junos OS Evolved, you create a template with the name `template_1`, create an inline-monitoring instance `i1`, and create the configuration for the on-box collector `c2`. For an on-box collector, you specify the `controller re` statement instead of a local destination address and port:

```
user@host# set services inline-monitoring instance i1 template-name template_1
user@host# set services inline-monitoring instance i1 collector c2 source-address 10.11.12.1
user@host# set services inline-monitoring instance i1 controller re
```

3. Configure the observation cloud identifier.

   An observation cloud is the largest set of observation domains. According to RFC 5101, an observation domain is the largest set of observation points for which flow information can be aggregated by a metering process. For example, a router line card may be an observation domain if it is composed of several interfaces, each of which is an observation point. By configuring an observation cloud, you allow inline-monitoring services to report on a set of common properties that is locally unique per observation cloud. For more information about observation clouds, see *inline-monitoring*. To configure the observation cloud identifier:

```
user@host# set services inline-monitoring observation-cloud-id identifier
```

In this example, you have configured the identifier as `1`:

```
user@host# set services inline-monitoring observation-cloud-id 1
```

4. Subscribe to various exception types and configure exception reporting for a particular PFE and specify the inline-monitoring instance. For Junos OS, you must specify a particular exception category name, such as `forwarding-state`. For Junos OS Evolved, you simply specify `all` as the category name.

   By default, the exception data is sent to an off-box collector. To configure:

   ```
   user@host# set chassis fpc slot-number pfe identifier exception-reporting category category-name inline-monitoring-instance inline-monitoring-instance-name
   ```

   For Junos OS:

   In this example, you subscribe to forwarding exceptions and configure FPC 0 to send forwarding exceptions to the inline-monitoring instance `i1`:

   ```
   user@host# set chassis fpc 0 pfe 0 exception-reporting category forwarding-state inline-monitoring-instance i1
   ```

   For Junos OS Evolved:

   In this example, you subscribe to all exception categories and configure FPC 0 to send exceptions to the inline-monitoring instance `i1`:

   ```
   user@host# set chassis fpc 0 pfe 0 exception-reporting category all inline-monitoring-instance i1
   ```

5. (Optional) Additionally, if you prefer to use the on-box collector instead of sending the data to an off-box collector, then configure an on-box storage location for the exception data.

   To configure:

   ```
   user@host# set system resiliency exceptions forwarding
   user@host# set system resiliency store fwding-file file-name
   user@host# set system resiliency store fwding-file size file-size
   ```

In this example, you configure the file in which to store the forwarding exception data:

```
user@host# set system resiliency exceptions forwarding
user@host# set system resiliency store fwding-file file1
user@host# set system resiliency store fwding-file size 1g
```

Exception Code Reference

SUMMARY

This section contains information about the exception codes and their explanations, for both Junos OS and Junos OS Evolved.

IN THIS SECTION

MX Exception Codes, Junos OS

To subscribe to a particular exception category, configure the category for a particular FPC and Packet Forwarding Engine. The command completion help shows the available categories:

```
[edit]
user@host# set chassis fpc 0 pfe 0 exception-reporting category ?
Possible completions:
  firewall             Firewall exceptions
  forwarding-state     Forwarding state related exceptions
  layer2               Layer 2 exceptions
  layer3               Layer 3 exceptions
  packet-errors        Packet format error related exceptions
```

In the following tables, there is a column with code values; for example, `73`. This value is the exception code that is reported. From an external viewpoint, when the exception is reported, the actual code sent is 128 + *value*. For example, the code value `73` would mean that for the `unknown family discard` exception reason, the code reported would be 128 + 73 = 201.

There is only one exception to this rule, the exception reason `ttl expired`, code value `1`. This reason is reported with the same code value; that is, 128 is not added to the value.

The following tables, organized by exception category, contain the exception reasons, codes, and code descriptions for the MX Series routers.

Table 63: Firewall Exception Category Codes, MX Series

Reason	Code	Description
firewall discard	67	Firewall in the packet path has a action.
firewall discard V6	101	Firewall in the IPv6 packet path has a action.
firewall discard out	113	Firewall in the output packet path has a action.
firewall discard V6 out	114	Firewall in the output IPv6 packet path has a `discard` action.

Table 64: Forwarding-State Exception Category Codes, MX Series

Reason	Code	Description
unknown family	73	Packets discarded because the corresponding address family is not configured on the interface.
unknown vrf	77	Routing instance table was not created properly for the VRF ID associated with the input interface.

Table 65: Layer 2 Exception Category Codes, MX Series

Reason	Code	Description
vlan id out of oif's range	32	Packet's VLAN ID is outside the VLAN range configured on the output interface.

Table 66: Layer 3 Exception Category Codes, MX Series

Reason	Code	Description
discard route	66	Discard nexthops explicitly installed by routing protocols.

Table 66: Layer 3 Exception Category Codes, MX Series *(Continued)*

Reason	Code	Description
discard route IPv6	102	Discard nexthops explicitly installed by IPv6 routing protocol.
hold route	70	Packet is discarded because it hit a hold nexthop.

Table 67: Packet-Errors Exception Category Codes, MX Series

Reason	Code	Description
bad ipv4 hdr checksum	2	IPv4 checksum verification has failed.
non-IPv4 layer3 tunnel	4	GRE/IPIP/PIM tunnels are supported only over IPv4, not IPv6.
GRE unsupported flags	5	Only the `Key Present Flag` is supported when de-encapsulating packets received with a GRE header.
tunnel pkt too short	6	Length computation error when de-encapsulating packets received with a tunnel header.
bad IPv6 options pkt	9	IPv6 packet includes too many (or too long) option headers. Option headers extend beyond byte 256 in the input parcel.
bad IPv4 hdr	11	IPv4 packet header error: • Version is not 4, or • Internet Header Length (IHL) is less than 5, or • IHL > Total length / 4.
bad IPv6 hdr	110	IPv6 packet header error. Version is not 6.

Table 67: Packet-Errors Exception Category Codes, MX Series *(Continued)*

Reason	Code	Description
bad IPv4 pkt len	12	IPv4 frame length too short (is less than Layer 2 encapsulated length plus Layer 3 length).
bad IPv6 pkt len	111	IPv6 frame length too short (is less than Layer 2 encapsulated length plus Layer 3 length).
mtu exceeded	21	A packet is discarded as its length, after new header imposition, exceeds the MTU configured on the logical interface. For IPv4/IPv6 packets, this indicates the `don't fragment` bit has been set in the IPv4 header for a packet that requires fragmentation.
ttl expired	1	Time To Live (TTL) has expired for the packet.
my-mac check failed	28	For Layer 3 interfaces, the `my-mac` check failed for these packets; hence are dropped.
my-mac check failed IPv6	63	For Layer 3 interfaces, the IPv6 `my-mac` check failed for these packets; hence are dropped.

PTX Series Exception Codes, Junos OS Evolved

Table 68 on page 410 contains the trap code numbers, the exception codes, and their descriptions for the PTX10004, PTX10008, and PTX10016 routers with the JNP10K-1201 and JNP10K-1202 line cards, and the PTX10001-36MR and PTX10003 routers.

For routers running Junos OS Evolved, there is only one exception category, `all`. For example, to subscribe to category `all` and configure FPC 0 to send exceptions to the inline-monitoring instance `i1`:

```
user@host# set chassis fpc 0 pfe 0 exception-reporting category all inline-monitoring-instance i1
```

Table 68: PTX Series Exception Codes, Junos OS Evolved

Trap Code	Exception Code	Description
1	dlu.ucode.inv_start_pc	A valid instruction to process packet was not found; a lookup failure. (For example: my MAC miss on Layer 3 interface.)
4	dlu.ucode.discard	DLU UCODE discard (lookup failure)
5	dlu.ucode.invalid_seq	DLU UCODE invalid sequence
6	dlu.ucode.ip_bc_with_my_mac	DLU UCODE IP broadcast with my MAC
7	dlu.ucode.unreachable	DLU UCODE unreachable
8	dlu.ucode.not_routable	DLU UCODE not routable
13	dlu.ucode.my_ll_mc	DLU UCODE my link-level multicast
14	dlu.ucode.bad_sip	DLU UCODE bad source IP address
15	dlu.ucode.ttl_exp	DLU UCODE TTL expired
16	dlu.ucode.oam_to_cpu	DLU UCODE OAM packet to CPU
17	dlu.ucode.ip_mc_iif_mismatch	DLU UCODE IP multicast interface index mismatch
18	dlu.ucode.ip_mc_resolve	DLU UCODE IP multicast resolve
19	dlu.ucode.vlan_tag_lookup_miss	DLU UCODE VLAN tag lookup miss
20	dlu.ucode.vtag_normalize_miss	DLU UCODE VLAN tag normalization miss
25	dlu.ipipe.err.trapcode.hw_err	Parity/ECC error
26	dlu.ipipe.err.trapcode.config_err	Route configuration error
27	dlu.ipipe.err.trapcode.proc_ttl_err	Route programming error (loop)

Table 68: PTX Series Exception Codes, Junos OS Evolved *(Continued)*

Trap Code	Exception Code	Description
29	dlu.ipipe.err.trapcode.mpls_buf_uflow	Lookup is beyond the supported label stack depth.
30	dlu.ipipe.err.trapcode.l3offs_oflow	Incorrect parsing of Layer 3 offset
31	dlu.ipipe.err.trapcode.seq_rsvd	Route programming error
34	dlu.ilp.lookup.err.trapcode.cfg_err	Route programming error
36	igp_misc.trapcode.l2l_invalidopt	Incorrectly constructed host injected packet
37	igp_misc.trapcode.invalid_dft_code	Incorrectly constructed host injected packet
38	igp_misc.trapcode.cpu_ple	Incorrectly constructed host injected packet
40	slu.trapcode.l2_domain_lookup_failure	Wrong type of packet (tagged/untagged) on the interface
41	slu.trapcode.stp_blocked	Packet on STP blocked port
44	slu.trapcode.eth_mcast	Source MAC address in the packet is multicast
45	slu.trapcode.eth_bcast	Source MAC address in the packet is broadcast
46	slu.trapcode.eth_src_eq_dest	SLU Ethernet - source address == destination address
47	slu.trapcode.l3offset	Incorrect packet (Layer 2 header size > 64)
48	slu.trapcode.v4_trunc_pkt	ipv4.totalLength < ipv4.ihl * 4
49	slu.trapcode.v4_ver	IPV4 illegal version
50	slu.trapcode.v4_mcast	SLU IPv4—multicast source IP address
51	slu.trapcode.v4_bcast	SLU IPv4—broadcast source IP address

Table 68: PTX Series Exception Codes, Junos OS Evolved *(Continued)*

Trap Code	Exception Code	Description
52	slu.trapcode.v4_src_eq_dest	SLU IPv4—source address == destination address
53	slu.trapcode.v4_size	SLU IPv4 minimum/maximum packet size check
54	slu.trapcode.v6_ver	SLU IPv6—illegal version
55	slu.trapcode.v6_src_eq_dest	SLU IPv6 - source address == destination address
56	slu.trapcode.v6_pyldleneq0	SLU IPv6—header payload length field is equal to 0
57	slu.trapcode.v6_size	SLU IPv6 minimum/maximum packet size check
58	slu.trapcode.tcp_tiny_attack_frag0	SLU TCP—tiny TCP attack with `frag_off` = 0
59	slu.trapcode.tcp_tiny_attack_frag1	SLU TCP—tiny TCP attack with `frag_off` = 1
60	slu.trapcode.ip_tcp_pyldlen	SLU IP/TCP—illegal IP payload length
61	slu.trapcode.ip_udp_pyldlen	SLU UDP—illegal IP payload length
62	slu.trapcode.ip_icmp_pyldlen	SLU ICMP—illegal IP payload length
63	slu.trapcode.ip_igmp_pyldlen	SLU IGMP—illegal IP payload length
64	slu.trapcode.ip_pim_pyldlen	SLU PIM—illegal IP payload length
65	slu.trapcode.ip_sctp_pyldlen	SLU SCTP—illegal IP payload length
66	slu.trapcode.ip_gre_pyldlen	SLU GRE—illegal IP payload length
67	slu.trapcode.ip_ah_pyldlen	SLU AH—illegal IP payload length
68	slu.trapcode.packet_length_err	SLU Trapcode value for minimum size packets

Table 68: PTX Series Exception Codes, Junos OS Evolved *(Continued)*

Trap Code	Exception Code	Description
69	slu.trapcode.v4_opt_hdr	SLU IPv4 option header is NOT one of the acceptable types
70	slu.trapcode.v6_ext_hdr	SLU IPv6 extension header IS one of the trappable types
71	slu.trapcode.v4_hdr_len	SLU IPv4—illegal header length
72	slu.trapcode.ah_hdr_len	SLU AH—illegal header length field
73	slu.trapcode.eth_repl	SLU trapcode for ethertype parsing limit reached
74	slu.trapcode.eth_nomatch	SLU no match found in `KnownEthertypes` CAM
75	slu.trapcode.v4_chksum	SLU IPv4—checksum error
76	slu.trapcode.tcp_hdr_len	SLU IPv4—illegal TCP header length
77	slu.trapcode.l2_ple	Packet Layer 2 header size is greater than 128 bytes
78	slu.trapcode.eth_ple	SLU Ethertype parser limit exceeded
79	slu.trapcode.v4_ple	SLU IPv4 parser limit exceeded
80	slu.trapcode.v6_ple	SLU IPv6 parser limit exceeded
81	slu.trapcode.tcp_ple	SLU TCP parser limit exceeded
82	slu.trapcode.udp_ple	SLU UDP parser limit exceeded
83	slu.trapcode.icmp_ple	SLU ICMP parser limit exceeded
84	slu.trapcode.igmp_ple	SLU IGMP parser limit exceeded
85	slu.trapcode.pim_ple	SLU PIM parser limit exceeded
86	slu.trapcode.sctp_ple	SLU SCTP parser limit exceeded

Table 68: PTX Series Exception Codes, Junos OS Evolved *(Continued)*

Trap Code	Exception Code	Description
87	slu.trapcode.gre_ple	SLU GRE parser limit exceeded
88	slu.trapcode.ah_ple	SLU AH parser limit exceeded
89	slu.trapcode.gtp_ple	SLU GTP parser limit exceeded
90	slu.trapcode.vxlan_ple	SLU VXLAN parser limit exceeded
91	slu.trapcode.oam_ple	SLU Ethernet OAM parser limit exceeded
92	slu.trapcode.ptp_ple	SLU PTP parser limit exceeded
93	slu.trapcode.eth_invalidtag	Invalid Ethernet tag
95	slu.trapcode.oam_discard	Ethernet OAM bad configuration/incorrect packet
96	slu.trapcode.fttl_exp	SLU fabric TTL expired check
97	slu.trapcode.oam_bad_mac	Ethernet OAM incorrect packet
98	slu.trapcode.tunnel_ip_options	Tunnel termination of IP options packet
99	slu.trapcode.tunnel_ttl_expired	TTL expired on tunnel packet
101	slu.trapcode.gport_tcam_miss	Incorrect tunnel packet
105	irp.core.trapcode.mem_err	Parity/ECC error
106	irp.core.trapcode.cfg_err	Incorrect hardware nexthop programming
107	irp.core.trapcode.pttl_expire	Incorrect hardware nexthop programming
108	irp.core.trapcode.voq_ip_option_ip4	Virtual output queue (VOQ) IP option IPv4 packet. (NOT ERROR)
109	irp.core.trapcode.voq_ip_option_ip6	VOQ IP option IPv6 packet. (NOT ERROR)
110	irp.core.trapcode.trap_after_lkup	IRP VOQ trap after lookup (debugging)

Table 68: PTX Series Exception Codes, Junos OS Evolved *(Continued)*

Trap Code	Exception Code	Description
111	irp.core.trapcode.storm0	BUM traffic exceeded configured limits
112	irp.core.trapcode.storm1	BUM traffic exceeded configured limits
113	irp.core.trapcode.storm2	BUM traffic exceeded configured limits
114	irp.core.trapcode.storm3	BUM traffic exceeded configured limits
115	irp.core.trapcode.voqcalc	Incorrect hardware nexthop programming
116	irp.core.trapcode.act_stbyte	Incoming packet does not satisfy split horizon criteria.
117	irp.core.trapcode.act_srcid	Incoming packet does not satisfy split horizon criteria.
118	irp.core.trapcode.act_igportid	Incoming packet does not satisfy split horizon criteria.
119	irp.core.trapcode.act_gl2dom0	Incoming packet does not satisfy split horizon criteria.
120	irp.core.trapcode.act_gl2dom1	Incoming packet does not satisfy split horizon criteria.
121	irp.core.trapcode.act_ref	Multicast-only fast reroute (MoFRR) aging (NOT ERROR)
122	irp.core.trapcode.ttl_ig_ip6	IPv6 ingress TTL expired
123	irp.core.trapcode.ttl_ig_ip4	IPv4 ingress TTL expired
124	irp.core.trapcode.ttl_eg_ip6	IPv6 egress TTL expired
125	irp.core.trapcode.ttl_eg_ip4	IPv4 egress TTL expired
126	irp.core.trapcode.trap_all	IRP trap all parcels
127	irp.core.trapcode.packetlen	Incorrect incoming packet

Table 68: PTX Series Exception Codes, Junos OS Evolved *(Continued)*

Trap Code	Exception Code	Description
128	irp.core.trapcode.vlanparse	Incorrect incoming packet
129	irp.core.trapcode.vlandiscard	Incorrect incoming packet
130	irp.core.trapcode.opthead	Incorrect incoming packet
131	irp.core.trapcode.policer	Packet dropped because of policer action
132	epp.epc.cfg.common.trapcode.hw_err	Parity/ECC error
133	epp.epc.cfg.common.trapcode.ntlu_cfg	Debugging (NOT ERROR)
135	epp.epc.cfg.common.trapcode.ptp_enable_trap	Debugging (NOT ERROR)
136	epp.epc.cfg.common.trapcode.l2_tags_exceeded_trap	Incorrect incoming packet (Layer 2 tags exceeded 46 bytes)
137	epp.epc.cfg.common.trapcode.dual_hash_miss	Programming error (Ingress to egress token programming). Transient error; can be ignored.
139	epp.epc.cfg.common.trapcode.reorder_timedout_pkt	Hardware egress processing error
140	epp.epc.cfg.common.trapcode.qos_rewrite_enable_trap	Debugging (NOT ERROR)
141	epp.epc.cfg.common.trapcode.illegal_qix	Egress nexthop programming error
143	epp.epc.cfg.common.trapcode.eth_tme	EPP more than 9 Ethertypes
144	epp.epc.cfg.common.trapcode.eth_passbuf	EPP passthru parsing failed
145	epp.epc.cfg.common.trapcode.oam_ple	EPP Ethernet OAM parser limit exceeded
146	epp.epc.cfg.common.trapcode.oam_discard	EPP Ethernet OAM discard

Table 68: PTX Series Exception Codes, Junos OS Evolved *(Continued)*

Trap Code	Exception Code	Description
147	epp.epc.cfg.common.trapcode.oam_bad_mac	EPP Ethernet bad MAC address (incorrect incoming packet)
148	epp.epe.cfg.elu.trapcode.hw_err	Parity/ECC error
149	epp.epe.cfg.elu.trapcode.desc_addr_err	EPP error generating descriptor address (programming error)
150	epp.epe.cfg.elu.trapcode.pdct_drop	Egress policer drop
151	epp.epe.cfg.elu.trapcode.pdct_err	Parity/ECC error
152	epp.epe.cfg.elu.trapcode.l3_mtu_chk_fail	MTU check exceeded.
153	epp.epe.cfg.elu.trapcode.df_set_for_fragmentation	DF set on incoming packet
154	epp.epe.cfg.elu.trapcode.mpls_ovfl	Incorrect egress nexthop programming (> 8 label push)
155	epp.epe.cfg.elu.trapcode.ing_pfe_ttl_exp	Ingress TTL expired
156	epp.epe.cfg.elu.trapcode.egr_pfe_ttl_exp	Egress TTL expired
157	epp.epe.cfg.elu.trapcode.desc_prog_err	EPP illegal descriptor programming
158	epp.epe.cfg.elu.trapcode.l2plen_ofl	EPP Layer 2 length is too large
160	epp.epe.cfg.elu.trapcode.l3_plen_chk_fail	EPP Layer 3 length failed minimum and maximum length check
161	epp.epe.cfg.elu.trapcode.dmac_prog_err	EPP illegal destination MAC programming
162	epp.epe.cfg.elu.trapcode.smac_prog_err	EPP illegal source MAC programming
163	epp.epe.cfg.elu.trapcode.l3l4_err	Incorrect egress nexthop programming
164	epp.epe.cfg.elu.trapcode.invld_vlan_sel	Invalid combination of VLAN tags

Table 68: PTX Series Exception Codes, Junos OS Evolved *(Continued)*

Trap Code	Exception Code	Description
165	epp.epe.cfg.elu.trapcode.tunnel_data_invld	EPP tunnel buffer construction error (programming error)
166	epp.epe.cfg.elu.trapcode.hdr_seq_rsvd	Incorrect egress nexthop programming (tunnel/template/MPLS)
167	epp.epe.cfg.elu.trapcode.rewrite_tmp_nxt_type	EPP rewrite enabled but `tmpNxtType` unknown
168	epp.epe.cfg.elu.trapcode.rewrite_mpls_buf_ovfl	EPP rewrite MPLS buffer overflow
169	epp.epe.cfg.elu.trapcode.rewrite_newheader_size_exceeded	EPP new header legal size exceeded
170	epp.epe.cfg.elu.trapcode.opt_hdr_err	EPP invalid option header
171	epp.epe.cfg.elu.trapcode.fft_prog_err	EPP FFT programming error
172	epp.epe.cfg.elu.trapcode.pkt_chk.same_port	EPP packet check same port
173	epp.epe.cfg.elu.trapcode.pkt_chk.same_lport	EPP packet check same lport
174	epp.epe.cfg.elu.trapcode.pkt_chk.same_l2domain	EPP packet check same Layer 2 domain
175	epp.epe.cfg.elu.trapcode.pkt_chk.ip_redir	EPP packet check IP redirect
176	epp.epe.cfg.elu.trapcode.pkt_chk.same_gl2domain	EPP packet check same gl Layer 2 domain
177	epp.epe.cfg.elu.trapcode.pkt_chk.split_horizon	EPP packet check split horizon
178	epp.epe.cfg.elu.trapcode.pkt_chk.same_port_l2d	EPP packet check same port/Layer 2 domain

Table 68: PTX Series Exception Codes, Junos OS Evolved *(Continued)*

Trap Code	Exception Code	Description
179	epp.epe.cfg.elu.trapcode.pkt_chk.same_source_id	EPP packet check same source ID
181	epp.epe.cfg.elu.trapcode.pkt_chk.is_true	EPP packet check `is_true`
220	sw.igp_ai_rule_invalid	SW IGP AI rule invalid
221	sw.igp_ai_invalid_pattern	SW IGP AI invalid pattern
222	sw.igp_ui_no_tag_support	SW IGP UI rule no tag support
223	sw.egp_ui_no_tag_support	SW EGP UI rule no tag support
248	sw.egnh.cfg_trap	Egress nexthop descriptor trap to CPU
249	sw.egnh.cfg_discard	Egress nexthop descriptor discard
250	sw.egnh.cfg_pfh_trap	Discard nexthop
251	sw.irp_nh_discard_sample	Discard nexthop

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
23.4R1-EVO	Support for the Juniper Resiliency Interface (PTX10003 router)—Starting in Junos OS Evolved Release 23.4R1, you can use the Juniper Resiliency Interface (JRI) to detect, correlate, and mitigate exceptions.
22.2R1-EVO	Support for the Juniper Resiliency Interface (PTX10001-36MR, PTX10004, PTX10008, and PTX10016 routers with the JNP10K-LC1201 or JNP10K-LC1202 linecards)—Starting in Junos OS Evolved Release 22.2R1, you can use the Juniper Resiliency Interface (JRI) to detect, correlate, and mitigate exceptions.

21.2R1	Support for the Juniper Resiliency Interface (MX304, MX480, MX960, MX2010, MX2020 and vMX)—Starting in Junos OS Release 21.2R1, you can use our new Juniper Resiliency Interface (JRI) to detect, correlate, and mitigate exceptions. JRI extends the inline monitoring services feature with Juniper-specific IPFIX information elements (IEs) for exception data and introduces the concept of an Observation Cloud, which is a set of Observation Domains. You can send the IPFIX packets to either an on-box or an off-box collector.

RELATED DOCUMENTATION

Inline Monitoring Services Configuration

4

PART

Sampling and Discard Accounting Services

CHAPTER 12

Sampling Data Using Traffic Sampling and Discard Accounting

IN THIS CHAPTER

Configuring Traffic Sampling on MX, M and T Series Routers

IN THIS SECTION

Traffic sampling enables you to copy traffic to a Physical Interface Card (PIC) that performs flow accounting while the router forwards the packet to its original destination. You can configure the router to perform sampling in one of the following three locations:

- On the Routing Engine, using the sampled process. To select this method, use a filter (input or output) with a matching term that contains the `then sample` statement.

- On the Monitoring Services, Adaptive Services, or Multiservices PIC.
- On an inline data path without the need for a services Dense Port Concentrator (DPC). To do this inline active sampling, you define a sampling instance with specific properties. One Flexible PIC Concentrator (FPC) can support only one instance; for each instance, either services PIC-based sampling or inline sampling is supported per family. Inline sampling supports version 9 and IPFIX flow collection templates.

> **NOTE**: Routing Engine based sampling is not supported on VPN routing and forwarding (VRF) instances.

Configuring Firewall Filter for Traffic Sampling

To configure firewall filter for traffic sampling, you must perform the following tasks:

- Create a firewall filter to apply to the logical interfaces being sampled by including the `filter` statement at the `[edit firewall family *family-name*]` hierarchy level. In the filter `then` statement, you must specify the action modifier `sample` and the action `accept`.

```
filter filter-name {
    term term-name {
        then {
            sample;
            accept;
        }
    }
}
```

For more information about firewall filter actions and action modifiers, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

- Apply the filter to the interfaces on which you want to sample traffic by including the `address` and `filter` statements at the `[edit interfaces *interface-name* unit *logical-unit-number* family *family-name*]` hierarchy level:

```
address address {
}
filter {
    input filter-name;
}
```

The following prerequisites apply to M, MX, and T Series routers when you configure traffic sampling on interfaces and in firewall filters:

- If you configure a sample action in a firewall filter for an inet or inet6 family on an interface without configuring the forwarding-options settings, operational problems might occur if you also configure port mirroring or flow-tap functionalities. In such a scenario, all the packets that match the firewall filter are incorrectly sent to the service PIC.

- If you include the `then sample` statement at the `[edit firewall family inet filter *filter-name* term *term-name*]` hierarchy level to specify a sample action in a firewall filter for IPv4 packets, you must also include the `family inet` statement at the `[edit forwarding-options sampling]` hierarchy level or the `instance *instance-name* family inet` statement at the `[edit forwarding-options sampling]` hierarchy level. Similarly, if you include the then `sample` statement at the `[edit firewall family inet6 filter *filter-name* term *term-name*]` hierarchy level to specify a sample action in a firewall filter for IPv6 packets, you must also include `family inet6` statement at the `[edit forwarding-options sampling]` hierarchy level or the `instance *instance-name* family inet6` statement at the `[edit forwarding-options sampling]` hierarchy level. Otherwise, a commit error occurs when you attempt to commit the configuration.

- Also, if you configure traffic sampling on a logical interface by including the sampling input or sampling output statements at the `[edit interface *interface-name* unit *logical-unit-number*]` hierarchy level, you must also include the `family inet | inet6` statement at the `[edit forwarding-options sampling]` hierarchy level, or the `instance *instance-name* family inet | inet6` statement at the `[edit forwarding-options sampling]` hierarchy level.

Configuring Traffic Sampling on a Logical Interface

To configure traffic sampling on any logical interface, enable sampling and specify a non zero sampling rate by including the sampling statement at the `[edit forwarding-options]` hierarchy level:

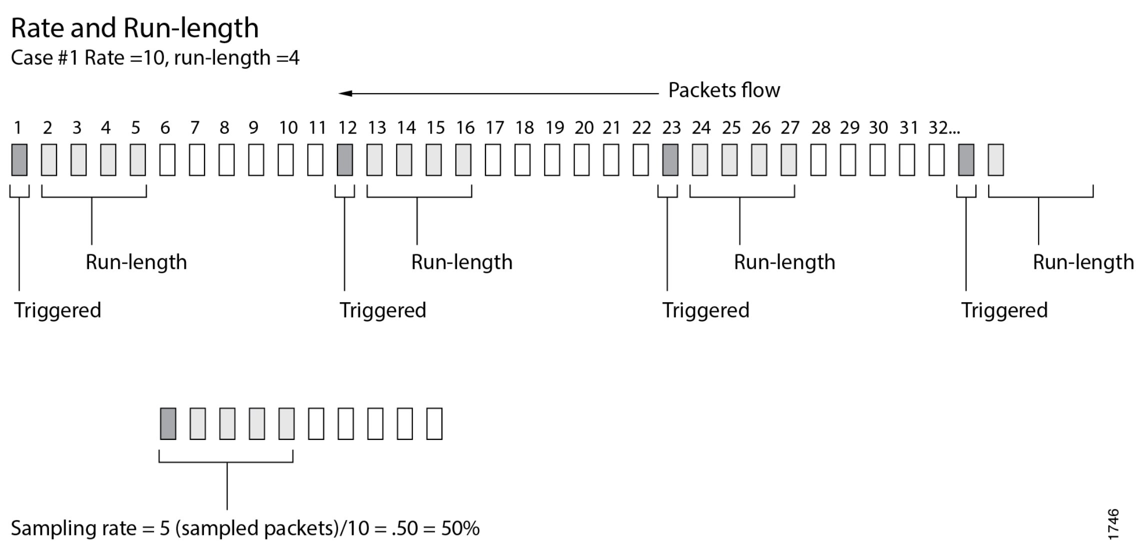
```
sampling {
    input {
        rate number;
        run-length number;
        max-packets-per-second number;
        maximum-packet-length bytes;
    }
```

When you use Routing Engine-based sampling, specify the threshold traffic value by including the `max-packets-per-second` statement. The value is the maximum number of packets to be sampled, beyond which the sampling mechanism begins dropping packets. The range is from 0 through 65,535. A value of 0 instructs the Packet Forwarding Engine not to sample any packets. The default value is 1000.

NOTE: When you configure active monitoring and specify a Monitoring Services, Adaptive Services, or Multiservices PIC in the `output` statement, or when you configure inline sampling, the `max-packets-per-second` value is ignored.

Specify the sampling rate by setting the values for `rate` and `run-length` (see Figure 54 on page 425).

Figure 54: Configuring Sampling Rate



NOTE: Do not configure ingress sampling on `ms-` logical interfaces on which PIC-based flow monitoring is enabled, which causes undesired flow monitoring behavior and might result in repeated sampling of a single packet. Starting in Junos OS Release 15.1, a commit error occurs when you try to configure ingress traffic sampling on that interface. In Junos OS Release 14.2 and earlier, the commit error does not occur, but you should not configure ingress traffic sampling on that interface.

If PIC-based flow monitoring is enabled on an ms-*fpc/pic/port.logical-unit* interface, a commit check error occurs when you attempt to configure ingress traffic sampling on that interface. This error occurs because a combination of ingress sampling and PIC-based flow monitoring operations on an `ms-` logical interface causes undesired flow monitoring behavior and might result in repeated sampling of a single packet. You must not configure ingress sampling on `ms-` logical interfaces on which PIC-based flow monitoring is enabled.

The `rate` statement specifies the ratio of packets to be sampled. For example, if you configure a rate of 10, *x* number of packets out of every 10 is sampled, where *x*=run length + 1. By default, the rate is 0, which means that no traffic is sampled.

The `run-length` statement specifies the number of matching packets to sample following the initial one-packet trigger event. By default, the run length is 0, which means that no more traffic is sampled after the trigger event. The range is from 0 through 20. Configuring a run length greater than 0 allows you to sample packets following those already being sampled.

> **NOTE**: The `run-length` and `maximum-packet-length` configuration statements are not supported on MX80 routers.

If you do not include the `input` statement, sampling is disabled.

To collect the sampled packets in a file, include the `file` statement at the `[edit forwarding-options sampling output]` hierarchy level. Output file formats are discussed later in the chapter.

Disabling Traffic Sampling

To explicitly disable traffic sampling on the router, include the `disable` statement at the `[edit forwarding-options sampling]` hierarchy level:

```
disable;
```

Sampling Once

To explicitly sample a packet for active monitoring only once, include the `sample-once` statement at the `[edit forwarding-options sampling]` hierarchy level:

```
sample-once;
```

Setting this option avoids duplication of packets in cases where sampling is enabled at both the ingress and egress interfaces and simplifies analysis of the sampled traffic.

Preserving Prerewrite ToS Value for Egress Sampled or Mirrored Packets

Starting in Junos OS Release 14.1, you can preserve the prenormalized type-of-service (ToS) value in egress sampled or mirrored packets. Include the `pre-rewrite-tos` statement at the `[edit forwarding-options sampling]` hierarchy level.

On MPC-based interfaces, you can configure ToS rewrite either using class-of-service (CoS) configuration by including the `rewrite-rules dscp` *`rule_name`* statement at the `[edit class-of-service interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level or using firewall filter configuration by including the `dscp` statement at the `[edit firewall family` *`family-name`* `filter` *`filter-name`* `term` *`term-name`* `then]` hierarchy level. If ToS rewrite is configured, the egress mirrored or sampled copies contain the post-rewrite ToS values by default. With the `pre-rewrite-tos` configuration, you can retain the prerewrite ToS value in the sampled or mirrored packets.

NOTE:

- If ToS rewrite is configured on the egress interface by using both CoS and firewall filter configuration, and if the `pre-rewrite-tos` statement is also configured, then the egress sampled packets contain the DSCP value set using the firewall filter configuration. However, if the `pre-rewrite-tos` statement is not configured, the egress sampled packets contain the DSCP value set by the CoS configuration.

- With the `pre-rewrite-tos` statement, you can configure retaining prenormalization ToS values only for sampling done under `family inet` and `family inet6`.

- This feature cannot be configured at the `[edit logical-systems]` hierarchy level. It can be configured only at the global level under the `forwarding-option` configuration.

- When ToS rewrite is configured by using a firewall filter on both ingress and egress interfaces, the egress sampled packets contain the DSCP value set by the ingress ToS rewrite configuration if the `pre-rewrite-tos` statement is configured. However, if the `pre-rewrite-tos` statement is not configured, the egress sampled packets contain the DSCP value set by the ToS rewrite configuration for the egress firewall filter.

- If the `pre-rewrite-tos` statement is configured, and a deactivate or delete operation is performed at the `[edit forwarding-options]` hierarchy level, `pre-rewrite-tos` configuration still remains active. To disable the `pre-rewrite-tos` configuration for such a case, you must explicitly deactivate or delete the `pre-rewrite-tos` statement at the `[edit forwarding-options sampling]` hierarchy level before performing a deactivate or delete operation at the `[edit forwarding-options]` hierarchy level.

Configuring Traffic Sampling Output

To configure traffic sampling output, include the following statements at the `[edit forwarding-options sampling family (inet | inet6 | mpls) output]` hierarchy level:

```
aggregate-export-interval seconds;
```

```
flow-active-timeout seconds;
flow-inactive-timeout seconds;
extension-service service-name;
    flow-server hostname {
        aggregation {
            autonomous-system;
            destination-prefix;
            protocol-port;
            source-destination-prefix {
                caida-compliant;
            }
            source-prefix;
        }
        autonomous-system-type (origin | peer);
        (local-dump | no-local-dump);
        port port-number;
        source-address address;
        version format;
        version9 {
            template template-name;
        }
    }
    interface interface-name {
        engine-id number;
        engine-type number;
        source-address address;
    }
    file {
        disable;
        filename filename;
        files number;
        size bytes;
        (stamp | no-stamp);
        (world-readable | no-world-readable);
    }
```

To configure inline flow monitoring on MX Series routers, include the `inline-jflow` statement at the `[edit forwarding-options sampling instance` *`instance-name`* `family (inet | inet6 | mpls) output]` hierarchy level. Inline sampling exclusively supports a new format called IP_FIX that uses UDP as the transport protocol. When you configure inline sampling, you must include the `version-ipfix` statement at the `[edit forwarding-options sampling instance` *`instance-name`* `family (inet | inet6 | mpls) output flow-server` *`address`*`]` hierarchy level and also at the `[edit services flow-monitoring]` hierarchy level. For more information about configuring inline flow monitoring, see "Configuring Inline Active Flow Monitoring Using Routers, Switches or NFX250" on page 74.

To direct sampled traffic to a flow-monitoring interface, include the `interface` statement. The `engine-id` and `engine-type` statements specify the identity and type numbers of the interface; they are dynamically generated based on the Flexible PIC Concentrator (FPC), PIC, and slot numbers and the chassis type. The `source-address` statement specifies the traffic source.

Starting in Junos OS Release 19.3R1, to configure inline flow monitoring on Juniper Advanced Threat Prevention Cloud (ATP Cloud), include the `flow-server` statement at the `[edit forwarding-options sampling instance` *`instance-name`* `family (inet | inet6 | mpls) output]` hierarchy level. Inline sampling exclusively supports a new format called IP_FIX that uses UDP as the transport protocol. When you configure inline sampling, you must include the `version-ipfix` statement at the `[edit forwarding-options sampling instance` *`instance-name`* `family (inet | inet6 | mpls) output flow-server` *`address`*`]` hierarchy level and also at the `[edit services flow-monitoring]` hierarchy level.

To configure flow sampling version 9 output, you need to include the `template` statement at the `[edit forwarding-options sampling output version9]` hierarchy level. For information on cflowd, see "Enabling Flow Aggregation" on page 595.

The `aggregate-export-interval` statement is described in "Configuring Discard Accounting" on page 438, and the `flow-active-timeout` and `flow-inactive-timeout` statements are described in "Configuring Flow Monitoring" on page 5.

Traffic sampling results are automatically saved to a file in the **/var/tmp** directory. To collect the sampled packets in a file, include the `file` statement at the `[edit forwarding-options sampling family inet output]` hierarchy level:

```
file {
    disable;
    filename filename;
    files number;
    size bytes;
    (stamp | no-stamp);
    (world-readable | no-world-readable);
}
```

Traffic Sampling Output Format

Traffic sampling output is saved to an ASCII text file. The following is an example of the traffic sampling output that is saved to a file in the **/var/tmp** directory. Each line in the output file contains information for one sampled packet. You can optionally display a timestamp for each line.

The column headers are repeated after each group of 1000 packets.

```
# Apr  7 15:48:50
Time                    Dest           Src Dest Src Proto TOS Pkt Intf  IP   TCP
                        addr          addr port port          len num frag flags
Apr 7 15:48:54 192.168.9.194 192.168.9.195   0    0   1   0x0  84  8   0x0   0x0
Apr 7 15:48:55 192.168.9.194 192.168.9.195   0    0   1   0x0  84  8   0x0   0x0
Apr 7 15:48:56 192.168.9.194 192.168.9.195   0    0   1   0x0  84  8   0x0   0x0
Apr 7 15:48:57 192.168.9.194 192.168.9.195   0    0   1   0x0  84  8   0x0   0x0
Apr 7 15:48:58 192.168.9.194 192.168.9.195   0    0   1   0x0  84  8   0x0   0x0
```

To set the timestamp option for the file `my-sample`, enter the following:

```
[edit forwarding-options sampling output file]
user@host# set filename my-sample files 5 size 2m world-readable stamp;
```

Whenever you toggle the timestamp option, a new header is included in the file. If you set the `stamp` option, the `Time` field is displayed.

```
# Apr  7 15:48:50
# Time            Dest        Src  Dest   Src Proto  TOS   Pkt  Intf    IP   TCP
#                 addr       addr  port  port              len   num  frag flags
# Feb  1 20:31:21
#                 Dest        Src  Dest   Src Proto  TOS   Pkt  Intf    IP   TCP
#                 addr       addr  port  port              len   num  frag flags
```

Tracing Traffic Sampling Operations

Tracing operations track all traffic sampling operations and record them in a log file in the **/var/log** directory. By default, this file is named **/var/log/sampled**. The default file size is 128K, and 10 files are created before the first one gets overwritten.

To trace traffic sampling operations, include the `traceoptions` statement at the `[edit forwarding-options sampling]` hierarchy level:

```
traceoptions {
    no-remote-trace;
    file filename <files number> <size bytes> <match expression> <world-readable | no-world-
```

```
readable>;
}
```

Traffic Sampling Examples

Example: Sampling a Single SONET/SDH Interface

The following configuration gathers statistical sampling information from a small percentage of all traffic on a single SONET/SDH interface and collects it in a file named `sonet-samples.txt`.

Create the filter:

```
[edit firewall family inet]
filter {
    input sample-sonet {
        then {
            sample;
            accept;
        }
    }
}
```

Apply the filter to the SONET/SDH interface:

```
[edit interfaces]
so-0/0/1 {
    unit 0 {
        family inet {
            filter {
                input sample-sonet;
            }
            address 10.127.68.254/32 {
                destination 172.16.74.7;
            }
        }
    }
}
```

Finally, configure traffic sampling:

```
[edit forwarding-options]
sampling {
    input {
        family inet {
            rate 100;
            run-length 2;
        }
    }
    family inet {
        output {
            file {
                filename sonet-samples.txt;
                files 40;
                size 5m;
            }
        }
    }
}
```

Example: Sampling All Traffic from a Single IP Address

The following configuration gathers statistical information about every packet entering the router on a specific Gigabit Ethernet port originating from a single source IP address of `172.16.92.31`, and collects it in a file named `samples-172-16-92-31.txt`.

Create the filter:

```
[edit firewall family inet]
filter one-ip {
    term get-ip {
        from {
            source-address 172.16.92.31;
        }
        then {
            sample;
            accept;
        }
```

```
    }
}
```

Apply the filter to the Gigabit Ethernet interface:

```
[edit interfaces]
ge-4/1/1 {
    unit 0 {
        family inet {
            filter {
                input one-ip;
            }
            address 10.45.92.254;
        }
    }
}
```

Finally, gather statistics on all the candidate samples; in this case, gather all statistics:

```
[edit forwarding-options]
sampling {
    input {
        family inet {
            rate 1;
        }
    }
    family inet {
        output {
            file {
                filename samples-172-16-92-31.txt;
                files 100;
                size 100k;
            }
        }
    }
}
```

Example: Sampling All FTP Traffic

The following configuration gathers statistical information about a moderate percentage of packets using the FTP data transfer protocol in the output path of a specific T3 interface, and collects the information in a file named `t3-ftp-traffic.txt`.

Create a filter:

```
[edit firewall family inet]
filter ftp-stats {
    term ftp-usage {
        from {
            destination-port [ftp ftp-data];
        }
        then {
            sample;
            accept;
        }
    }
}
```

Apply the filter to the T3 interface:

```
[edit interfaces]
t3-7/0/2 {
    unit 0 {
        family inet {
            filter {
                input ftp-stats;
            }
            address 10.35.78.254/32 {
                destination 10.35.78.4;
            }
        }
    }
}
```

Finally, gather statistics on 10 percent of the candidate samples:

```
[edit forwarding-options]
sampling {
```

```
    input {
        family inet {
            rate 10;
        }
    }
    family inet {
        output {
            file {
                filename t3-ftp-traffic.txt;
                files 50;
                size 1m;
            }
        }
    }
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can preserve the prenormalized type-of-service (ToS) value in egress sampled or mirrored packets. Include the `pre-rewrite-tos` statement at the `[edit forwarding-options sampling]` hierarchy level.

RELATED DOCUMENTATION

Traffic Sampling, Forwarding, and Monitoring Overview

Configuring Sampling Instance on MX, M and T Series Routers or QFX Series Switches | 435

Configuring Sampling Instance on MX, M and T Series Routers or QFX Series Switches

You can configure active sampling by defining a sampling instance that specifies a name for the sampling parameters and bind the instance name to an FPC, MPC, or DPC. This configuration enables you to define multiple named sampling parameter sets associated with multiple destinations and protocol families per sampling destination. With the cflowd version 5 and version 8 and flow aggregation version 9, you can use templates to organize the data gathered from sampling.

To implement this feature, you include the `instance` statement at the `[edit forwarding-options sampling]` hierarchy level.

The following considerations apply to the sampling instance configuration:

- This configuration is supported on the IP version 4 (`inet`), IP version 6 (`ipv6`), and MPLS protocol families.
- You can configure the router to perform sampling in either of two locations:
  - On the Routing Engine, using the sampled process. To select this method, use a filter (input or output) with a matching term that contains the `then` sample statement.
  - On the Monitoring Services, Adaptive Services, or Multiservices PIC. Specify the interface name at the `[forwarding-options sampling instance` *`instance-name`* `family inet output interface]` hierarchy level. You can configure the same or different services PICs in a set of sampling instances.
- You can configure the `rate` and `run-length` options at the `[edit forwarding-options sampling input]` hierarchy level to apply common values for all families on a global basis. Alternatively, you can configure these options at the `[edit forwarding-options sampling instance` *`instance-name`* `input]` hierarchy level to apply specific values for each instance or at the `[edit forwarding-options sampling instance` *`instance-name`* `family` *`family`* `input]` hierarchy level to apply specific values for each protocol family you configure.
- Starting in Junos OS Release 16.1, for inline active flow monitoring, you can configure a Differentiated Services Code Point (DSCP) mapping and a forwarding class to apply to exported packets. Use the `dscp` and `forwarding-class` options at the `[edit forwarding-options sampling` *`instance-name`* `family (inet | inet6) output flow-server` *`hostname`*`]` hierarchy level.
- For MX Series devices with Modular Port Concentrators (MPCs), port-mirrored or sampled packets can be truncated (or clipped) to any length in the range of 1 through 255 bytes. Only the values 1 to 255 are valid for packet truncation on these devices. For other devices, the range is from 0 through 9216. A maximum-packet-length value of zero (0) represents that truncation is disabled, and the entire packet is mirrored or sampled.

> **NOTE**: The `run-length` and `maximum-packet-length` configuration statements are not supported on MX80 routers.

To associate the defined instance with a particular FPC, MPC, or DPC, you include the `sampling-instance` statement at the `[edit chassis fpc *number*]` hierarchy level, as in the following example:

```
chassis {
    fpc 2 {
        sampling-instance samp1;
    }
}
```

Starting in Junos OS Release 14.1, you can associate a sampling instance with an FPC in the MX Series Virtual Chassis primary or backup router. Use the **sampling-instance *instance-name*** statement at the **[edit chassis member *member-number* fpc slot *slot-number*]** hierarchy level, where *member-number* is 0 (for the primary router) or 1 (for the backup router), and *slot-number* is a number in the range 0 through 11.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
16.1	Starting in Junos OS Release 16.1, for inline active flow monitoring, you can configure a Differentiated Services Code Point (DSCP) mapping and a forwarding class to apply to exported packets.
14.1	Starting in Junos OS Release 14.1, you can associate a sampling instance with an FPC in the MX Series Virtual Chassis primary or backup router.

RELATED DOCUMENTATION

Traffic Sampling, Forwarding, and Monitoring Overview

Monitoring, Sampling, and Collection Services Interfaces User Guide

Configuring Active Flow Monitoring | 42

Directing Traffic Sampling Output to a Server Running the cflowd Application

Configuring Traffic Sampling on MX, M and T Series Routers | 422

Example: Sampling Instance Configuration | 133

sampling (Forwarding Options)

Inline Flow Monitoring for Virtual Chassis Overview

Configuring Discard Accounting

Discard accounting is similar to traffic sampling, but varies from it in two ways:

- In discard accounting, the packet is intercepted by the monitoring PIC and is not forwarded to its destination.
- Traffic sampling allows you to limit the number of packets sampled by configuring the `max-packets-per-second`, `rate`, and `run-length` statements. Discard accounting does not provide these options, and a high packet count can potentially overwhelm the monitoring PIC.

A discard instance is a named entity that specifies collector information under the `accounting` *name* statement. Discard instances are referenced in firewall filter `term` statements by including the `then discard accounting` *name* statement.

Most of the other statements are also found at the `[edit forwarding-options sampling]` hierarchy level. For information on cflowd, see "Enabling Flow Aggregation" on page 595. The `flow-active-timeout` and `flow-inactive-timeout` statements are described in "Configuring Flow Monitoring" on page 5.

To direct sampled traffic to a flow-monitoring interface, include the `interface` statement. The `engine-id` and `engine-type` statements specify the accounting interface used on the traffic, and the `source-address` statement specifies the traffic source.

You cannot use rate-limiting with discard accounting; however, you can specify the duration of the interval for exporting aggregated accounting information by including the `aggregate-export-interval` statement in the configuration. This enables you to put a boundary on the amount of traffic exported to a flow-monitoring interface.

RELATED DOCUMENTATION

Enabling Flow Aggregation | 595

Configuring Flow Monitoring | 5

CHAPTER 13

Sampling Data Using Inline Sampling

IN THIS CHAPTER

Understand Inline Active Flow Monitoring

SUMMARY

Inline active flow monitoring is implemented on the Packet Forwarding Engine. The Packet Forwarding Engine performs functions such as creating flows, updating flows, and exporting flow records to a flow collector. The flow records are sent out in industry-standard IPFIX or version 9 format. IPFIX and version 9 templates use UDP as the transport protocol.

IN THIS SECTION

You can configure inline active flow monitoring for IPv4, IPv6, MPLS, MPLS-IPv4, VPLS, and bridge traffic. See the change history table at the end of this topic for details about particular platform support.

For PTX Series, starting with Junos OS Evolved Release 21.2R1 and Junos OS Release 21.3R1, no flows are maintained. For ACX Series running Junos OS Evolved, no flows are maintained either. Every sampled packet is considered to be a flow. When the sampled packet is received, the flow is created and immediately timed out as inactive, and the software exports a record to the collector. Therefore, the number of records sent to the collector is higher than before. See Table 69 on page 440. The IPFIX and version 9 Options Template Data Record now contains 0 in the `Flow Active Timeout` (Element ID 36) and `Flow Inactive Timeout` (Element ID 37) fields. Therefore, the Options Template Data Record is not compliant with IPFIX RFC 7011. The `show services accounting flow inline-jflow fpc-slot` *`slot`* operational mode command now displays 0 for all of the `Active Flows` and `Timed Out` fields. The various `Total Flows` fields are now equal to their respective `Flow Packets` fields. The various `Flows Inactive Timed Out` fields are now equal to their respective `Flow Packets` fields. The effect of the `nexthop-learning` statement at the `[edit services flow-monitoring version` *`version`* `template` *`template-name`*`]` hierarchy level on this no-flow behavior varies depending upon the operating system. For Junos OS Evolved, we do not recommend that you configure the `nexthop-learning` statement, as it reduces the number of packets that can be processed. For Junos OS, you can configure the `nexthop-learning` statement to change this default no-flow behavior and once again create and maintain flows, then attach the template to all sampling instances associated with FPCs that require the previous behavior.

Table 69: Inline Active Flow Monitoring Behavior Comparison for PTX Series

Actions	Prior to Junos OS Evolved Release 21.2R1 and Junos OS Release 21.3R1	Starting in Junos OS Evolved Release 21.2R1 and Junos OS Release 21.3R1
Flow creation	Flows are created and maintained.	No flows are created. Every packet is considered as a new flow for accounting purposes.
Active timeout	Active timeout configuration is honored. Active flows are timed out if the traffic is continuous. An export record is created for the timed-out flow and exported to the collector.	Active timeout configuration is ignored. No flows are timed out.
Inactive timeout	Inactive timeout configuration is honored. Inactive flows are timed out and are deleted at that time. An export record is created for the timed-out flow and exported to the collector.	Inactive timeout configuration is ignored. All flows are inactively timed out immediately.

Table 69: Inline Active Flow Monitoring Behavior Comparison for PTX Series *(Continued)*

Actions	Prior to Junos OS Evolved Release 21.2R1 and Junos OS Release 21.3R1	Starting in Junos OS Evolved Release 21.2R1 and Junos OS Release 21.3R1
Export records creation	Export records are created only during timeouts.	Export records are created for every sampled packet.
Packet export to collector	The configured active and Inactive timeouts determine the packet export rates to the collector.	The packet export rate to the collector is directly proportional to sampling rate (in packets per second) at that given point in time. Because each packet results in an export record, the number of packets sent out to the collector does increase in comparison to what it was before.

Benefits of Inline Active Flow Monitoring

Inline active flow monitoring is implemented on the Packet Forwarding Engine rather than on a services card. This enables:

- Lower cost—You do not need to invest in additional hardware.
- Higher scalability—You do not need to dedicate a PIC slot for a services PIC, so you can make full use of the available slots for handling traffic on the device.
- Better performance—Inline flow monitoring performance is not dependent on the capacity of a services card.

Inline Active Flow Monitoring Configuration Overview

The inline active flow monitoring configuration can be broadly classified into four categories:

1. Configurations at the `[edit services flow-monitoring]` hierarchy level—At this level, you configure the template properties for inline flow monitoring.
2. Configurations at the `[edit forwarding-options]` hierarchy level—At this level, you configure a sampling instance and associate the template (configured at the `[edit services flow-monitoring]` hierarchy level) with the sampling instance. At this level, you also configure the flow-server IP address and port number as well as the flow export rate, and specify the collectors.

You cannot change the source IP address for collectors under the same family. Also, the template mapped across collectors under a family should be the same.

3. Configurations at the `[edit chassis]` hierarchy level—At this level, you associate the sampling instance with the FPC on which the media interface is present. If you are configuring sampling of IPv4 flows, IPv6 flows, or VPLS flows, you can configure the flow hash table size for each family.

4. Configurations at the `[edit firewall]` hierarchy level—At this level you configure a firewall filter for the family of traffic to be sampled. You must attach this filter to the interface on which you want to sample the traffic.

Inline Active Flow Monitoring Optional Features

We also support the following optional features:

Non-default VPN Routing and Forwarding (VRF) and mgmt_junos VRF Instances

Non-default VRF instance: For Junos OS, you can configure IPFIX and version 9 flow collectors to be reachable through WAN ports belonging to non-default VRF instances by including the `routing-instance` *`instance-name`* statement at the `[edit forwarding-options sampling instance` *`instance-name`* `family (inet |inet6 | mpls) output flow-server` *`hostname`*`]` hierarchy level for inline flow monitoring. For Junos OS Evolved, starting in Release 24.2R1 for supported platforms, you can configure IPFIX and version 9 flow collectors to be reachable through WAN ports belonging to non-default VRF instances. In prior releases, you could only configure flow collectors that were reachable through the default VRF instance. You cannot configure a flow collector to be reachable through non-default VRF instances for version 5 and version 8 flows. You must also configure the routing instance to be a VRF instance by including the `instance-type vrf` statement at the `[edit routing-instances instance-name]` hierarchy level. Use Feature Explorer to confirm platform and release support for this feature for Junos OS Evolved: Export of IPFIX and version 9 records of sampled packets through the management interface and WAN ports belonging to the non-default VRF instance.

mgmt_junos VRF instance: Starting in Junos OS Evolved 24.2R1 for supported platforms, you can export IPFIX or version 9 sampled packets to collectors reachable through interfaces belonging to the mgmt_junos VRF instance. You can configure collectors for both types of VRF instances in the same sampling configuration. However, the collectors reachable through the mgmt_junos VRF instance and the collectors reachable through the WAN ports cannot coexist under the same family, because you can specify only one source IP address per family. You can specify `inet` collectors, `inet6` collectors, or a mix of the two types. Use Feature Explorer to confirm platform and release support for this feature for Junos OS Evolved: Export of IPFIX and version 9 records of sampled packets through the management interface and WAN ports belonging to the non-default VRF instance.

Track Dropped Traffic

Starting in Junos OS Release 15.1F7 for certain MX Series routers, you can configure the `report-zero-oif-gw-on-discard` at the `[edit chassis fpc slot-number inline-services]` hierarchy level to set the flow record value to 0 (zero) for the IPFIX information elements `ipNextHopIPv6Address` (Element ID 62), `egressInterface` (Element ID 14), and `ipNextHopIPv4Address` (Element ID 15) for only the flow records for rejected, discarded, and policed flows. You configure this statement to help you track what traffic is being dropped.

Accurate BGP Next-Hop Addresses

Starting in Junos OS Release 24.2R1 and in Junos OS Evolved Release 24.4R1, we have added support for reporting an accurate BGP next-hop address for traffic load-balanced over multiple BGP peers in the ingress direction. Prior to this release, we reported the first in a list of BGP next hops. To contain this accurate BGP next-hop address, we use the IPv4 BGP Nexthop Address (IE 18) field in the IPv4 and MPLS-IPv4 templates and the IPv6 BGP Nexthop Address (IE 63) field in the IPv6 and MPLS-IPv6 templates, for both the IPFIX and the version 9 formats. For Junos OS only, to configure this feature, include the `multi-bgp-path` statement at the `[edit services flow-monitoring (version-ipfix | version9) template]` hierarchy level. You do not need to configure this statement for Junos OS Evolved.

For this feature to work for both Junos OS and Junos OS Evolved, you must configure the `nexthop-learning enable` statement at the `[edit services flow-monitoring version version template template-name]` hierarchy level.

For Junos OS only, for IPv6 and MPLS-IPv6 flows, you also need to configure the `ipv6-extended-attrib` statement at the `[edit chassis fpc slot-number inline-services flow-table-size]` hierarchy level. You do not need to configure this statement for Junos OS Evolved. When this feature is enabled, the `fragmentIdentification` (IE 54) field reports a value of 0. For limitations and caveats for this feature, see multi-bgp-path. Use Feature Explorer to confirm platform and release support for this feature: Inline active flow monitoring multiple BGP next hop support.

Support for a Demux0 Interface Mapped to an Underlying Aggregated Ethernet (AE) Interface

Starting in Junos OS Release 24.2R1, we support configuring inline active flow monitoring for a demux0 interface mapped to an underlying AE interface, for core-facing interfaces only. We support only the mpls-template of family MPLS and for protocol family CCC. (No ip4-template or ipv6-template support.) Existing maximum flow scale on the linecard is supported. Demux0 is a single interface under which you can map an AE interface, which in turn hosts multiple child logical interfaces. The flow record reports the SNMP ID of the underlying AE logical interface as the egress interface. Demux0 logical interfaces are not supported on the reserved unit numbers 16383 and 32767; for example, demux0.16383 and demux0.32767 are not supported. Use Feature Explorer to confirm platform and release support for this feature: Inline active flow monitoring on demux interfaces.

You configure the MPLS template and sampling instance as you normally would. Additionally, you configure the demux0 and the AE interfaces and the firewall filter to sample the traffic:

```
set interfaces xe-1/0/0:0 gigether-options 802.3ad ae1
set interfaces ae1 encapsulation flexible-ethernet-services
set interfaces ae1 unit 0 encapsulation vlan-ccc
set interfaces ae1 unit 0 vlan-tags outer 400
set interfaces ae1 unit 0 vlan-tags inner 100
set interfaces ae1 unit 0 input-vlan-map pop-swap
set interfaces ae1 unit 0 input-vlan-map inner-vlan-id 100
set interfaces ae1 unit 0 output-vlan-map swap-push
set interfaces ae1 unit 0 family ccc
set interfaces ae1 unit 0 family mpls filter input mpls_filter
set interfaces demux0 unit 0 encapsulation vlan-ccc
set interfaces demux0 unit 0 vlan-tags outer 400
set interfaces demux0 unit 0 vlan-tags inner 100
set interfaces demux0 unit 0 input-vlan-map pop-swap
set interfaces demux0 unit 0 input-vlan-map inner-vlan-id 100
set interfaces demux0 unit 0 output-vlan-map swap-push
set interfaces demux0 unit 0 demux-options underlying-interface ae1
set interfaces demux0 unit 0 family ccc
set interfaces demux0 unit 0 family inet address 10.1.1.1/24
set firewall family mpls filter mpls_filter term t1 then count mpls_c
set firewall family mpls filter mpls_filter term t1 then sample
set firewall family mpls filter mpls_filter term t1 then accept
```

Inline Active Flow Monitoring Limitations and Restrictions

The following limitations and restrictions apply to the inline active flow monitoring feature:

- Before you configure inline active flow monitoring, ensure that you have adequately-sized hash tables for IPv4, IPv6, MPLS, and VPLS flow sampling. (VPLS flow sampling is Junos OS only.) These tables can use from one up to fifteen 256K areas. Starting with Junos OS Release 16.1R1 and 15.1F2, the IPv4 table is assigned a default value of 1024. Prior to Junos OS Release 16.1 and 15.1F2, the IPv4 table is assigned a default value of fifteen 256K areas. The IPv6 table is assigned a default value of 1024, and the VPLS table is assigned a default value of 1024. Allocate larger tables when anticipated traffic volume makes it necessary.

- Inline active flow monitoring is not supported for input or output traffic on MS-MPC or MS-MIC-16G interfaces.

- Configuring both sFlow and inline active flow monitoring on the same interface leads to unexpected behavior. Therefore, configure these features on separate interfaces.
- Configuring both egress port mirroring and inline active flow monitoring on the same interface leads to unexpected behavior. Therefore, configure these features on separate interfaces.
- Ingress and egress sampling are sent to the same host-path queue. The packet rate in the queue is shared across ingress and egress sampled packets.
- Forwarding class configuration is not effective. Export record packets are always considered to be control frames and as such are pushed to the network-control queue.
- If multiple inline active flow monitoring firewall filters match to a flow, only the actions of the first filter are taken.
- In ingress sampling, if the destination port is on an aggregated Ethernet interface, the output interface is invalid.
- In Junos OS release 15.1 and earlier, you can apply version 9 flow templates to IPv4 traffic. Starting in Junos OS Release 16.1, you can also apply version 9 flow templates to MPLS and MPLS-IPv4 traffic. Starting in Junos OS Release 18.1R1, you can also apply version 9 flow templates to IPv6 traffic.
- In Junos OS Release 15.1 and earlier, you can apply IPFIX flow templates to IPv4, IPv6, and VPLS traffic. Starting in Junos OS release 16.1, you can also apply IPFIX flow templates to MPLS and MPLS-IPv4 traffic.
- Starting with Junos OS Release 17.2R1, you can apply IPFIX flow templates to unicast IPv4 and IPv6 traffic on QFX10002 switches. Starting with Junos OS Release 17.4R1, you can apply IPFIX flow templates to unicast IPv4 and IPv6 traffic on QFX10008 and QFX10016 switches.
- Inline active flow monitoring is not supported when you enable Next Gen Services on an MX Series router.
- You can configure only one sampling instance on a Flexible PIC Concentrator (FPC). For the ACX7509, you can configure a sampling instance only on FPC0.
- For Junos OS Evolved ACX routers only, a maximum of 40,000 samples/second is supported. Sampled packets have an additional FTMH header that causes an increase in the packet size.
- You can configure only one type of sampling—either services-card-based sampling or inline sampling—per family in a sampling instance. However, you can configure services-card-based and inline sampling for different families in a sampling instance.
- The following considerations apply to the inline sampling instance configuration:
  - Sampling run-length and clip-size are not supported.

- In Junos OS Release 16.2 and in Junos OS Release 16.1R3 and earlier, you can configure only one collector under a family for inline active flow monitoring. Starting with Junos OS Release 16.1R4 and 17.2R1, you can configure up to four collectors under a family for inline active flow monitoring. Starting with Junos OS Evolved 20.3R1, for the PTX10003 and PTX10008 (with the JNP10K-LC1201 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring. Starting with Junos OS Evolved 20.4R1, for the PTX10001-36MR and the PTX10008 (with the JNP10K-LC1202 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring. Starting with Junos OS Evolved 21.1R1, for the PTX10004 router, you can configure up to four collectors for inline active flow monitoring. To configure a collector under a family for inline active flow monitoring, configure the `flow-server` statement at the `edit forwarding-options sampling-instance` *`instance-name`* `family (inet | inet6) output` hierarchy level. To specify up to four collectors, include up to four `flow-server` statements.

- The user-defined sampling instance gets precedence over the global instance. When a user-defined sampling instance is attached to the FPC, the global instance is removed from the FPC and the user-defined sampling instance is applied to the FPC.

- Flow records and templates cannot be exported if the flow collector is reachable through any management interface, except as noted in "Non-default VPN Routing and Forwarding (VRF) and mgmt_junos VRF Instances " on page 442.

- If the destination of the sampled flow is reachable through multiple paths, the IP_NEXT_HOP (Element ID 15) and OUTPUT_SNMP (Element ID 14) in the IPv4 and IPv6 flow records are not reported correctly unless you enable learning of next hop addresses by using the `nexthop-learning enable` statement. (Starting in Junos OS Evolved Release 21.2R1 for PTX Series, we do not recommend that you enable learning of next-hop addresses, as it reduces the number of packets that can be processed. However, starting in Junos OS Release 21.3R1 for PTX Series, you can configure the `nexthop-learning` statement to change the default no-flow behavior and once again create and maintain flows, then attach the template to all sampling instances associated with FPCs that require the previous behavior.) If you do *not* configure `nexthop-learning enable`:

  - For IPv4 flow records, the IP_NEXT_HOP and OUTPUT_SNMP are set to the Gateway Address and SNMP Index of the first path seen in the forwarding table.

  - For IPv6 flow records, the IP_NEXT_HOP and OUTPUT_SNMP are set to 0.

- The Incoming Interface (IIF) and Outgoing Interface (OIF) should be part of the same VRF. If OIF is in a different VRF, DST_MASK (Element ID 13), DST_AS (Element ID 17), IP_NEXT_HOP (Element ID 15), and OUTPUT_SNMP (Element ID 14) are set to 0 in the flow records.

- Each lookup chip maintains and exports flows independent of other lookup chips. Traffic received on a media interface is distributed across all lookup chips in a multi-lookup chip platform. It is likely that a single flow is processed by multiple lookup chips. Therefore, each lookup chip creates a unique flow and exports it to the flow collector. This can cause duplicate flow records to go to the flow collector.

The flow collector should aggregate PKTS_COUNT and BYTES_COUNT for duplicate flow records to derive a single flow record.

IPFIX and Version 9 Templates

Fields Included in the IPFIX Bridge Template for MX Series

Table 70 on page 447 shows the fields that are included in the IPFIX Bridge template. The fields are shown in the order in which they appear in the template.

Table 70: IPFIX Bridge Template Fields for MX, M, and T Series

Field	Element ID
Destination MAC	80
Source MAC	56
Ethernet Type	256
Input SNMP	10
Output SNMP	14
Flow End Reason	136
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Fields Included in the IPFIX IPv4 Template for MX, M, and T Series

Table 71 on page 448 shows the fields that are included in the IPFIX IPv4 template. The fields are shown in the order in which they appear in the template.

Table 71: IPFIX IPv4 Template Fields for MX, M, and T Series

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 ToS	5
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10
VLAN ID	58
IPv4 Source Mask	9
IPv4 Destination Mask	13
Source AS	16
Destination AS	17

Table 71: IPFIX IPv4 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
IPv4 Next Hop Address	15
TCP Flags	6
Output Interface	14
Minimum TTL	52
Maximum TTL	53
Flow End Reason	136
IP Protocol Version	60
BGP IPv4 Next Hop Address	18
Flow Direction (Starting in Junos OS Release 16.1)	61
802.1Q VLAN identifier (dot1qVlanId)	243
802.1Q Customer VLAN identifier (dot1qCustomerVlanId)	245
IP Identifier	54
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to Epoch time	152

Table 71: IPFIX IPv4 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
Time the flow ended with respect to Epoch time	153

Fields Included in the IPFIX IPv4 Template for PTX3000 Series, PTX5000 Series, and the PTX10001-20C Router

Table 72 on page 450 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 72: IPFIX IPv4 Template Fields for PTX3000 Series, PTX5000 Series, and the PTX10001-20C Router

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 TOS	5
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10
Source AS	16

Table 72: IPFIX IPv4 Template Fields for PTX3000 Series, PTX5000 Series, and the PTX10001-20C Router *(Continued)*

Field	Element ID
Destination AS	17
BGP IPv4 Next Hop Address	18
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv4 Next Hop Address	15
IPv4 Source Mask	9
IPv4 Destination Mask	13
TCP Flags	6
IP Protocol Version	60
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Table 72: IPFIX IPv4 Template Fields for PTX3000 Series, PTX5000 Series, and the PTX10001-20C Router *(Continued)*

Field	Element ID
The type of interface where packets are being received. This field can have the following values: • 1—Other (default value) • 131—De-encapsulated GRE traffic is reported as *tunnel*	368

Fields Included in the IPFIX IPv4 Template for PTX1000, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), and PTX10016 Series

Table 73 on page 452 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 73: IPFIX IPv4 Template Fields for PTX1000, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), and PTX10016 Series

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 TOS	5
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10

Table 73: IPFIX IPv4 Template Fields for PTX1000, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), and PTX10016 Series *(Continued)*

Field	Element ID
Source AS	16
Destination AS	17
BGP IPv4 Next Hop Address	18
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv4 Next Hop Address	15
IPv4 Source Mask	9
IPv4 Destination Mask	13
TCP Flags	6
IP Protocol Version	60
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Table 73: IPFIX IPv4 Template Fields for PTX1000, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), and PTX10016 Series *(Continued)*

Field	Element ID
The type of interface where packets are being received. This field can have the following values: • 1—Other (default value) • 131—De-encapsulated GRE traffic is reported as *tunnel*	368
Forwarding Class Name (first two bytes)	32767
Packet Loss Priority; this field can have the following values: • 0x00: Low • 0x01: Medium-low • 0x02: Medium-high • 0x03: High • 0xFF: Unknown	32766

Fields Included in the IPFIX IPv4 Template for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers

Table 74 on page 454 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 74: IPFIX IPv4 Template Fields for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers

Field	Element ID
IPv4 Source Address	8

Table 74: IPFIX IPv4 Template Fields for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers *(Continued)*

Field	Element ID
IPv4 Destination Address	12
IPv4 TOS	5
IPv4 Protocol	4
Source Port	7
Destination Port	11
ICMP Type and Code	32
Input SNMP Index	10
Source AS	16
Destination AS	17
IPv4 BGP Next Hop Address	18
IPv6 BGP Next Hop Address (PTX10002-36QDD only)	63
Output SNMP Index	14
Number of Bytes	1
Number of Packets	2
Time the flow started with respect to system up time (FPC up time)	22

Table 74: IPFIX IPv4 Template Fields for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers *(Continued)*

Field	Element ID
Time the flow ended with respect to system up time (FPC up time)	21
IPv4 Next Hop	15
IPv4 Source Mask	9
IPv4 Destination Mask	13
TCP Flags	6
IP Protocol Version	60
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153
The type of interface where packets are being received. This field can have the following values: • 1—Other (default value) • 131—De-encapsulated GRE traffic is reported as *tunnel*	368
Source AS Path List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	16 (list of this type)
Destination AS Path List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	17 (list of this type)

Table 74: IPFIX IPv4 Template Fields for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers *(Continued)*

Field	Element ID
BGP Source Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	484
BGP Destination Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	485
BGP Source Extended Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	487
BGP Destination Extended Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	488
BGP Source Large Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	490
BGP Destination Large Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	491
Forwarding Class Name (first two bytes)	32767

Table 74: IPFIX IPv4 Template Fields for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers *(Continued)*

Field	Element ID
Packet Loss Priority; this field can have the following values: • 0x00: Low • 0x01: Medium-low • 0x02: Medium-high • 0x03: High • 0xFF: Unknown	32766
Policy Map Name	32765 (PTX10002-36QDD only)
Minimum TTL	52
Maximum TTL	53

Fields Included in the IPFIX IPv4 Template for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100 Routers

Table 75 on page 458 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 75: IPFIX IPv4 Template Fields for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100 Routers

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12

Table 75: IPFIX IPv4 Template Fields for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100 Routers ***(Continued)***

Field	Element ID
IPv4 TOS	5
IPv4 Protocol	4
Source Port	7
Destination Port	11
ICMP Type and Code	32
Input SNMP Index	10
Source AS	16
Destination AS	17
BGP Next Hop Address	18
Output SNMP Index	14
Number of Bytes	1
Number of Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv4 Next Hop	15

Table 75: IPFIX IPv4 Template Fields for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100 Routers *(Continued)*

Field	Element ID
IPv4 Source Mask	9
IPv4 Destination Mask	13
TCP Flags	6
IP Protocol Version	60
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153
The type of interface where packets are being received. This field can have the following values: • 1—Other (default value) • 131—De-encapsulated GRE traffic is reported as *tunnel*	368
Forwarding Class Name (first two bytes)	32767
Packet Loss Priority; this field can have the following values: • 0x00: Low • 0x01: Medium-low • 0x02: Medium-high • 0x03: High • 0xFF: Unknown	32766

Fields Included in the IPFIX IPv6 Template for MX, M, and T Series

Table 76 on page 461 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 76: IPFIX IPv6 Template Fields for MX, M, and T Series

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 ToS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	139
Input Interface	10
VLAN ID	58
IPv6 Source Mask	29
IPv6 Destination Mask	30
Source AS	16
Destination AS	17

Table 76: IPFIX IPv6 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
IPv6 Next Hop Address	62
IPv6 BGP Next Hop Address	63
TCP Flags	6
Output Interface	14
Minimum Hop Limits	52
Maximum Hop Limits	53
Flow End Reason	136
Flow Direction (Starting in Junos OS Release 16.1)	61
802.1Q VLAN identifier (dot1qVlanId)	243
802.1Q Customer VLAN identifier (dot1qCustomerVlanId)	245
IP Identifier	54
IPv6 Option Headers	64
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to Epoch time	152

Table 76: IPFIX IPv6 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
Time the flow ended with respect to Epoch time	153

Fields Included in the IPFIX IPv6 Template for PTX3000 Series, PTX5000 Series, and the PTX10001-20C Router

Table 77 on page 463 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 77: IPFIX IPv6 Template Fields for PTX3000 Series, PTX5000 Series, and the PTX10001-20C Router

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 TOS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code (IPv6)	139
Input Interface	10
IPv6 Source Mask	29

Table 77: IPFIX IPv6 Template Fields for PTX3000 Series, PTX5000 Series, and the PTX10001-20C Router *(Continued)*

Field	Element ID
IPv6 Destination Mask	30
TCP Flags	6
Source AS	16
Destination AS	17
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv6 Next Hop Address	62
IPv6 BGP NextHop Address	63
IP Protocol Version	60
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Table 77: IPFIX IPv6 Template Fields for PTX3000 Series, PTX5000 Series, and the PTX10001-20C Router *(Continued)*

Field	Element ID
The type of interface where packets are being received. This field can have the following values: • 1—Other (default value) • 131—De-encapsulated GRE traffic is reported as *tunnel*	368

Fields Included in the IPFIX IPv6 Template for PTX1000, PTX10002-60C, QFX10002-60C,PTX10008 (without the JNP10008-SF3), and PTX10016 Series

Table 78 on page 465 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 78: IPFIX IPv6 Template Fields for PTX1000, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), and PTX10016 Series

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 TOS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code (IPv6)	139
Input Interface	10

Table 78: IPFIX IPv6 Template Fields for PTX1000, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), and PTX10016 Series *(Continued)*

Field	Element ID
IPv6 Source Mask	29
IPv6 Destination Mask	30
TCP Flags	6
Source AS	16
Destination AS	17
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv6 Next Hop Address	62
IPv6 BGP Next Hop Address	63
IP Protocol Version	60
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Table 78: IPFIX IPv6 Template Fields for PTX1000, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), and PTX10016 Series *(Continued)*

Field	Element ID
The type of interface where packets are being received. This field can have the following values: • 1—Other (default value) • 131—De-encapsulated GRE traffic is reported as *tunnel*	368
Forwarding Class Name (first two bytes)	32767
Packet Loss Priority; this field can have the following values: • 0x00: Low • 0x01: Medium-low • 0x02: Medium-high • 0x03: High • 0xFF: Unknown	32766

Fields Included in the IPFIX IPv6 Template for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers

Table 79 on page 467 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 79: IPFIX IPv6 Template Fields for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers

Field	Element ID
IPv6 Source Address	27

Table 79: IPFIX IPv6 Template Fields for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers *(Continued)*

Field	Element ID
IPv6 Destination Address	28
IPv6 TOS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code (IPv6)	139
Input SNMP Index	10
Source AS	16
Destination AS	17
IPv6 BGP Next Hop Address	63
Output SNMP Index	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21

Table 79: IPFIX IPv6 Template Fields for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers *(Continued)*

Field	Element ID
IPv6 Next Hop Address	62
IPv6 Source Mask	29
IPv6 Destination Mask	30
TCP Flags	6
IP Protocol Version	60
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153
The type of interface where packets are being received. This field can have the following values: • 1—Other (default value) • 131—De-encapsulated GRE traffic is reported as *tunnel*	368
Source AS Path List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	16 (list of this type)
Destination AS Path List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	17 (list of this type)
BGP Source Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	484

Table 79: IPFIX IPv6 Template Fields for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers *(Continued)*

Field	Element ID
BGP Destination Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	485
BGP Source Extended Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	487
BGP Destination Extended Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	488
BGP Source Large Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	490
BGP Destination Large Community List (when configured on the data-record-fields statement at the [edit services flow-monitoring version-ipfix template template-name] hierarchy level)	491
Forwarding Class Name (first two bytes)	32767
Packet Loss Priority; this field can have the following values: • 0x00: Low • 0x01: Medium-low • 0x02: Medium-high • 0x03: High • 0xFF: Unknown	32766

Table 79: IPFIX IPv6 Template Fields for PTX10001-36MR, PTX10002-36QDD, PTX10003-160C, PTX10003-80C, PTX10004, and PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3) Routers *(Continued)*

Field	Element ID
Policy Map Name	32765 (PTX10002-36QDD only)
Minimum TTL	52
Maximum TTL	53

Fields Included in the IPFIX IPv6 Template for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100

Table 80 on page 471 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 80: IPFIX IPv6 Template Fields for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100 Routers

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 TOS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11

Table 80: IPFIX IPv6 Template Fields for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100 Routers *(Continued)*

Field	Element ID
ICMP Type and Code (IPv6)	139
Input Interface	10
IPv6 Source Mask	29
IPv6 Destination Mask	30
TCP Flags	6
IP Protocol Version	60
Source AS	16
Destination AS	17
IPv6 BGP Next Hop Address	63
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv6 Next Hop Address	62

Table 80: IPFIX IPv6 Template Fields for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100 Routers *(Continued)*

Field	Element ID
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153
The type of interface where packets are being received. This field can have the following values: • 1—Other (default value) • 131—De-encapsulated GRE traffic is reported as *tunnel*	368
Forwarding Class Name (first two bytes)	32767
Packet Loss Priority; this field can have the following values: • 0x00: Low • 0x01: Medium-low • 0x02: Medium-high • 0x03: High • 0xFF: Unknown	32766

Fields Included in the IPFIX MPLS-IPv4 Template for MX, M, and T Series

Starting in Junos OS Release 16.1, the IPFIX MPLS-IPv4 template is supported. Table 81 on page 473 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 81: IPFIX MPLS-IPv4 Template Fields for MX, M, and T Series

Field	Element ID
MPLS Label 1	70

Table 81: IPFIX MPLS-IPv4 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
MPLS Label 2	71
MPLS Label 3	72
MPLS Top Label IP Address	47
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 ToS	5
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10
VLAN ID	58
IPv4 Source Mask	9
IPv4 Destination Mask	13
Source AS	16

Table 81: IPFIX MPLS-IPv4 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
Destination AS	17
IPv4 Next Hop Address	15
TCP Flags	6
Output Interface	14
Minimum TTL	52
Maximum TTL	53
Flow End Reason	136
IP Protocol Version	60
BGP IPv4 Next Hop Address	18
Flow Direction	61
802.1Q VLAN identifier (dot1qVlanId)	243
802.1Q Customer VLAN identifier (dot1qCustomerVlanId)	245
IP Identifier	54
Number of Flow Bytes	1
Number of Flow Packets	2

Table 81: IPFIX MPLS-IPv4 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Fields Included in the IPFIX MPLS-IPv4 Template for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), and PTX10016 Series

Starting in Junos OS Release 18.2R1, the IPFIX MPLS-IPv4 template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS-IPv4 template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS-IPv4 template is supported for the QFX10002-60C switch.Table 82 on page 476 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 82: IPFIX MPLS-IPv4 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 ToS	5
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32

Table 82: IPFIX MPLS-IPv4 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series *(Continued)*

Field	Element ID
Input Interface	10
Source AS	16
Destination AS	17
BGP IPv4 Next Hop Address	18
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153
IPv4 Next Hop Address	15
IPv4 Source Mask	9
IPv4 Destination Mask	13
TCP Flags	6

Table 82: IPFIX MPLS-IPv4 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series *(Continued)*

Field	Element ID
IP Protocol Version	60
Ingress Interface Type	368
MPLS Label 1	70
MPLS Label 2	71
MPLS Label 3	72
MPLS Top Label IPv6 Address	140

Fields Included in the IPFIX MPLS-IPv4 Template for PTX10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD.

Table 83 on page 478 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 83: IPFIX MPLS-IPv4 Template Fields for PTX Series, for PTX10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD.

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 ToS	5

Table 83: IPFIX MPLS-IPv4 Template Fields for PTX Series, for PTX10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD. ***(Continued)***

Field	Element ID
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10
Source AS	16
Destination AS	17
BGP IPv4 Next Hop Address	18
IPv6 BGP Next Hop Address (PTX10002-36QDD only)	63
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
Time the flow started with respect to Epoch time	152

Table 83: IPFIX MPLS-IPv4 Template Fields for PTX Series, for PTX10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD. ***(Continued)***

Field	Element ID
Time the flow ended with respect to Epoch time	153
IPv4 Next Hop Address	15
IPv4 Source Mask	9
IPv4 Destination Mask	13
TCP Flags	6
IP Protocol Version	60
MPLS Label 1	70
MPLS Label 2	71
MPLS Label 3	72
MPLS Top Label IPv6 Address	140
Minimum TTL	52
Maximum TTL	53

Fields Included in the IPFIX MPLS-IPv6 Template for MX, M, and T Series

Starting in Junos OS Release 18.4R1, the IPFIX MPLS-IPv6 template is supported for the MX Series. Table 84 on page 481 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 84: IPFIX MPLS-IPv6 Template Fields for MX, M, and T Series

Field	Element ID
MPLS Label 1	70
MPLS Label 2	71
MPLS Label 3	72
MPLS Top Label IP Address (Only IPv4 top label addresses are exported. IPv6 top label addresses report a value of zero.)	47
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 ToS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code (IPv6)	139
Input Interface	10
VLAN ID	58
IPv6 Source Mask	29
IPv6 Destination Mask	30

Table 84: IPFIX MPLS-IPv6 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
Source AS	16
Destination AS	17
IPv6 Next Hop Address	62
IPv6 BGP Next Hop Address	63
TCP Flags	6
Output Interface	14
Minimum TTL	52
Maximum TTL	53
Flow End Reason	136
Flow Direction	61
802.1Q VLAN identifier (dot1qVlanId)	243
802.1Q Customer VLAN identifier (dot1qCustomerVlanId)	245
IP Identifier	54
IPv6 Option Headers	64
Number of Flow Bytes	1

Table 84: IPFIX MPLS-IPv6 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
Number of Flow Packets	2
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Fields Included in the IPFIX MPLS-IPv6 Template for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series

Starting in Junos OS Release 18.2R1, the IPFIX MPLS-IPv6 template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS-IPv6 template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS-IPv6 template is supported for the QFX10002-60C switch. Table 85 on page 483 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 85: IPFIX MPLS-IPv6 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 ToS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11

Table 85: IPFIX MPLS-IPv6 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series *(Continued)*

Field	Element ID
ICMP Type and Code (IPv6)	139
Input Interface	10
Source AS	16
Destination AS	17
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153
IPv6 Source Mask	29
IPv6 Destination Mask	30
IPv6 Next Hop Address	62

Table 85: IPFIX MPLS-IPv6 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series *(Continued)*

Field	Element ID
IPv6 BGP Next Hop Address	63
TCP Flags	6
IP protocol version of IP payload on MPLS VPN	60
Ingress Interface Type	368
RSVP label (top MPLS label stack entry) for MPLS tunnel	70
RSVP label pushed before top label	71
MPLS Label 3	72
MPLS Top Label IPv6 Address	140

Fields Included in the IPFIX MPLS-IPv6 Template for PTX10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD.

Table 86 on page 485 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 86: IPFIX MPLS-IPv6 Template Fields for PTX10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, PTX10002-36QDD.

Field	Element ID
IPv6 Source Address	27

Table 86: IPFIX MPLS-IPv6 Template Fields for PTX10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, PTX10002-36QDD. *(Continued)*

Field	Element ID
IPv6 Destination Address	28
IPv6 ToS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code (IPv6)	139
Input Interface	10
Source AS	16
Destination AS	17
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
Time the flow started with respect to Epoch time	152

Table 86: IPFIX MPLS-IPv6 Template Fields for PTX10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, PTX10002-36QDD. *(Continued)*

Field	Element ID
Time the flow ended with respect to Epoch time	153
IPv6 Source Mask	29
IPv6 Destination Mask	30
IPv6 Next Hop Address	62
IPv6 BGP Next Hop Address	63
TCP Flags	6
IP protocol version of IP payload on MPLS VPN	60
RSVP label (top MPLS label stack entry) for MPLS tunnel	70
RSVP label pushed before top label	71
MPLS Label 3	72
MPLS Top Label IPv6 Address	140
Minimum TTL	52
Maximum TTL	53

Fields Included in the IPFIX MPLS Template for MX, M, and T Series

Starting in Junos OS Release 16.1, the IPFIX MPLS template is supported. Table 87 on page 488 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 87: IPFIX MPLS Template Fields for MX, M, and T Series

Field	Element ID
MPLS Label 1	70
MPLS Label 2	71
MPLS Label 3	72
Input Interface	10
Output Interface	14
Flow End Reason	136
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Fields Included in the IPFIX MPLS Template for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series

Starting in Junos OS Release 18.2R1, the IPFIX MPLS template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS template is supported for the QFX10002-60C switch. Table 88 on page 489 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 88: IPFIX MPLS Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series

Field	Element ID
Input Interface	10
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153
Ingress Interface Type	368
MPLS Label 1	70
MPLS Label 2	71
MPLS Label 3	72

Fields Included in the IPFIX MPLS Template for PTX10003, PTX10004, PTX10008, and PTX10016 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD.

Table 89 on page 490 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 89: IPFIX MPLS Template Fields for PTX10003, PTX10004, PTX10008 and PTX10016 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD.

Field	Element ID
Input Interface	10
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153
MPLS Label 1	70
MPLS Label 2	71
MPLS Label 3	72

Fields Included in the IPFIX MPLS-over-UDP Template for PTX Series and the QFX10002-60C Switch for Flows Within an IP Network and Having an IPv4 Payload

Starting in Junos OS Release 18.1R1, the IPFIX MPLS-over-UDP template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS-over-UDP template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS-over-UDP template is supported for the QFX10002-60C switch.

Inline active flow monitoring for MPLS-over-UDP traffic is not supported on the PTX10001-36MR, PTX10002-36QDD, PTX10003, PTX10004, and PTX10008 (with the JNP10008-SF3) routers.

Table 90 on page 491 shows the fields that are available in the IPFIX template for MPLS-over-UDP flows that are within an IP network and have an IPv4 payload. The fields are shown in the order in which they appear in the template.

Table 90: IPFIX MPLS-over-UDP Carried on IP Network Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch

Field	Element ID
IPv4 source address for tunnel endpoint	8
IPv4 destination address for tunnel endpoint	12
UDP source port for tunnel endpoint	7
Tunnel endpoint destination transport port	11
IPv4 source mask for tunnel source IP address	9
IPv4 destination mask for tunnel destination IP address	13
Source AS for tunnel	16
Destination AS for tunnel	17
IPv4 next hop address—gateway for tunnel destination IP address	15

Table 90: IPFIX MPLS-over-UDP Carried on IP Network Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
BGP IPv4 next hop address—tunnel destination IP BGP peer	18
Input SNMP index	10
Output SNMP index	14
MPLS label 1—VPN bottom of stack label	70
IP protocol version of IP payload on MPLS VPN	60
IPv4 source address of tunnel payload	8
IPv4 destination address of tunnel payload	12
IP protocol of tunnel payload	4
IP TOS	5
Source transport port	7
Destination transport port	11
ICMP type	32
TCP flags	6
Number of flow bytes	1
Number of flow packets	2

Table 90: IPFIX MPLS-over-UDP Carried on IP Network Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Fields Included in the IPFIX MPLS-over-UDP Template for PTX Series and the QFX10002-60C Switch for Flows Encapsulated in an RSVP-TE LSP and Having an IPv4 Payload

Starting in Junos OS Release 18.1R1, the IPFIX MPLS-over-UDP template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS-over-UDP template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS-over-UDP template is supported for the QFX10002-60C switch.

Inline active flow monitoring for MPLS-over-UDP traffic is not supported on the PTX10001-36MR, PTX10002-36QDD, PTX10003, PTX10004, and PTX10008 (with the JNP10008-SF3) routers.

Table 91 on page 493 shows the fields that are available in the IPFIX template for MPLS-over-UDP flows that are encapsulated in an RSVP-TE LSP in the inner MPLS network and have an IPv4 payload. The fields are shown in the order in which they appear in the template.

Table 91: IPFIX MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch

Field	Element ID
RSVP label (top MPLS label stack entry) for MPLS tunnel	70
RSVP label pushed before top label	71

Table 91: IPFIX MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
IPv4 source address for tunnel endpoint	8
IPv4 destination address for tunnel endpoint	12
UDP source port for tunnel endpoint	7
Tunnel endpoint destination transport port	11
IPv4 source mask for tunnel source IP address	9
IPv4 destination mask for tunnel destination IP address	13
Source AS for tunnel	16
Destination AS for tunnel	17
IPv4 next hop address—gateway for tunnel destination IP address	15
BGP IPv4 next hop address—tunnel destination IP BGP peer	18
Input SNMP index	10
Output SNMP index	14
MPLS label 1—VPN bottom of stack label	70
IP protocol version of IP payload on MPLS VPN	60
IPv4 source address of tunnel payload	8

Table 91: IPFIX MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
IPv4 destination address of tunnel payload	12
IP protocol of tunnel payload	4
IP TOS	5
Source transport port	7
Destination transport port	11
ICMP type	32
TCP flags	6
Number of flow bytes	1
Number of flow packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Fields Included in the IPFIX MPLS-over-UDP Template for PTX Series and the QFX10002-60C Switch for Flows Within an IP Network Having an IPv6 Payload

Starting in Junos OS Release 18.1R1, the IPFIX MPLS-Over-UDP template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS-over-UDP template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS-over-UDP template is supported for the QFX10002-60C switch.

Inline active flow monitoring for MPLS-over-UDP traffic is not supported on the PTX10001-36MR, PTX10002-36QDD, PTX10003, PTX10004, and PTX10008 (with the JNP10008-SF3) routers.

Table 92 on page 496 shows the fields that are available in the IPFIX template for MPLS-over-UDP flows that are within an IP network and have an IPv6 payload. The fields are shown in the order in which they appear in the template.

Table 92: IPFIX MPLS-over-UDP Carried on IP Network Template Fields (IPv6 Payload) for PTX Series and the QFX10002-60C Switch

Field	Element ID
IPv4 source address for tunnel endpoint	8
IPv4 destination address for tunnel endpoint	12
UDP source port for tunnel endpoint	7
Tunnel endpoint destination transport port	11
IPv4 source mask for tunnel source IP address	9
IPv4 destination mask for tunnel destination IP address	13
Source AS for tunnel	16
Destination AS for tunnel	17
IPv4 next hop address—gateway for tunnel destination IP address	15

Table 92: IPFIX MPLS-over-UDP Carried on IP Network Template Fields (IPv6 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
BGP next hop address—tunnel destination IP BGP peer	18
Input SNMP index	10
Output SNMP index	14
MPLS label 1—VPN bottom of stack label	70
IP protocol version of IP payload on MPLS VPN	60
IPv6 source address of tunnel payload	27
IPv6 destination address of tunnel payload	28
IP protocol of tunnel payload	4
IP TOS	5
Source transport port	7
Destination transport port	11
ICMP type V6	139
TCP flags	6
Number of flow bytes	1
Number of flow packets	2

Table 92: IPFIX MPLS-over-UDP Carried on IP Network Template Fields (IPv6 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Fields Included in the IPFIX MPLS-over-UDP Template for PTX Series and the QFX10002-60C Switch for Flows Encapsulated in an RSVP-TE LSP and Having an IPv6 Payload

Starting in Junos OS Release 18.1R1, the IPFIX MPLS-Over-UDP template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS-over-UDP template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS-over-UDP template is supported for the QFX10002-60C switch.

Inline active flow monitoring for MPLS-over-UDP traffic is not supported on the PTX10001-36MR, PTX10002-36QDD, PTX10003, PTX10004, and PTX10008 (with the JNP10008-SF3) routers.

Table 93 on page 498 shows the fields that are available in the IPFIX template for MPLS-over-UDP flows that are encapsulated in an RSVP-TE LSP in the inner MPLS network and have an IPv6 payload. The fields are shown in the order in which they appear in the template.

Table 93: IPFIX MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv6 Payload) for PTX Series and the QFX10002-60C Switch

Field	Element ID
RSVP label (top MPLS label stack entry) for MPLS tunnel	70
RSVP label pushed before top label	71

Table 93: IPFIX MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv6 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
IPv4 source address for tunnel endpoint	8
IPv4 destination address for tunnel endpoint	12
UDP source port for tunnel endpoint	7
Tunnel endpoint destination transport port	11
IPv4 source mask for tunnel source IP address	9
IPv4 destination mask for tunnel destination IP address	13
Source AS for tunnel	16
Destination AS for tunnel	17
IPv4 next hop address—gateway for tunnel destination IP address	15
BGP next hop address—tunnel destination IP BGP peer	18
Input SNMP index	10
Output SNMP index	14
MPLS label 1—VPN bottom of stack label	70
IP protocol version of IP payload on MPLS VPN	60
IPv6 source address of tunnel payload	27

Table 93: IPFIX MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv6 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
IPv6 destination address of tunnel payload	28
IP protocol of tunnel payload	4
IP TOS	5
Source transport port	7
Destination transport port	11
ICMP type V6	139
TCP flags	6
Number of flow bytes	1
Number of flow packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Fields Included in the IPFIX VPLS Template for MX, M, and T Series

Starting in Junos OS Release 16.1, the IPFIX VPLS template is supported. Table 94 on page 501 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 94: IPFIX VPLS Template Fields for MX, M, and T Series

Field	Element ID
Destination MAC	80
Source MAC	56
Ethernet Type	256
Input Interface	10
Output Interface	14
Flow End Reason	136
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to Epoch time	152
Time the flow ended with respect to Epoch time	153

Fields Included in the Version 9 Bridge Template for MX Series

Table 95 on page 501 shows the fields that are included in the version 9 Bridge template. The fields are shown in the order in which they appear in the template.

Table 95: Version 9 Bridge Template Fields for MX

Field	Element ID
Destination MAC	80

Table 95: Version 9 Bridge Template Fields for MX *(Continued)*

Field	Element ID
Source MAC	56
Ethernet Type	256
Input SNMP	10
Output SNMP	14
Flow End Reason	136
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to Epoch time	22
Time the flow ended with respect to Epoch time	21

Fields Included in the Version 9 IPv4 Template for MX, M, and T Series

Table 96 on page 502 shows the fields that are included in the version 9 IPv4 template. The fields are shown in the order in which they appear in the template.

Table 96: Version 9 IPv4 Template Fields for MX, M, and T Series

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12

Table 96: Version 9 IPv4 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
IPv4 ToS	5
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10
VLAN ID	58
IPv4 Source Mask	9
IPv4 Destination Mask	13
Source AS	16
Destination AS	17
IPv4 Next Hop Address	15
TCP Flags	6
Output Interface	14
Minimum TTL	52

Table 96: Version 9 IPv4 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
Maximum TTL	53
Flow End Reason	136
Internet Protocol Version	60
BGP IPv4 Next Hop Address	18
Flow Direction	61
802.1Q VLAN identifier (dot1qVlanId)	243
802.1Q Customer VLAN identifier (dot1qCustomerVlanId)	245
IP Identifier	54
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21

Fields Included in the Version 9 IPv4 Template for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series

Table 97 on page 505 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 97: Version 9 IPv4 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 TOS	5
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10
Source AS	16
Destination AS	17
BGP IPv4 Next Hop Address	18
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22

Table 97: Version 9 IPv4 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series *(Continued)*

Field	Element ID
Time the flow ended with respect to system up time (FPC up time)	21
IPv4 Next Hop Address	15
IPv4 Source Mask	9
IPv4 Destination Mask	13
TCP Flags	6
IP Protocol Version	60

Fields Included in the Version 9 IPv4 Template for PTX10003-160C, PTX10003-80C, PTX10004, PTX10008 (with the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD Routers

Table 98 on page 506 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 98: Version 9 IPv4 Template Fields for PTX10003-160C, PTX10003-80C, PTX10004, PTX10008 (with the JNP10008-SF3), PTX10001-36MR, PTX10002-36QDD Routers

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 TOS	5

Table 98: Version 9 IPv4 Template Fields for PTX10003-160C, PTX10003-80C, PTX10004, PTX10008 (with the JNP10008-SF3), PTX10001-36MR, PTX10002-36QDD Routers *(Continued)*

Field	Element ID
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input SNMP Index	10
IPv4 Source Mask	9
IPv4 Destination Mask	13
Source AS	16
Destination AS	17
IPv4 Next Hop Address	15
BGP IPv4 Next Hop Address	18
IPv6 BGP Next Hop Address (PTX10002-36QDD only)	63
TCP Flags	6
Number of Flow Bytes	1
Number of Flow Packets	2

Table 98: Version 9 IPv4 Template Fields for PTX10003-160C, PTX10003-80C, PTX10004, PTX10008 (with the JNP10008-SF3), PTX10001-36MR, PTX10002-36QDD Routers *(Continued)*

Field	Element ID
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IP Protocol Version	60
Output SNMP Index	14
Policy Map Name	32765 (PTX10002-36QDD only)
Minimum TTL	52
Maximum TTL	53

Fields Included in the Version 9 IPv4 Template for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100 Routers

Table 99 on page 508 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 99: Version 9 IPv4 Template Fields for ACX7024X, ACX7332, and ACX7348, ACX7509, and ACX7100 Routers

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 TOS	5

Table 99: Version 9 IPv4 Template Fields for ACX7024X, ACX7332, and ACX7348, ACX7509, and ACX7100 Routers *(Continued)*

Field	Element ID
IPv4 Protocol	4
Source Port	7
Destination Port	11
ICMP Type and Code	32
Input SNMP Index	10
Source AS	16
Destination AS	17
BGP Next Hop Address	18
Output SNMP Index	14
Number of Bytes	1
Number of Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv4 Next Hop	15
IPv4 Source Mask	9

Table 99: Version 9 IPv4 Template Fields for ACX7024X, ACX7332, and ACX7348, ACX7509, and ACX7100 Routers *(Continued)*

Field	Element ID
IPv4 Destination Mask	13
TCP Flags	6
IP Protocol Version	60

Fields Included in the Version 9 IPv6 Template for MX, M, and T Series

Starting in Junos OS Release 18.1R1, the version 9 IPv6 template is supported. Table 100 on page 510 shows the fields in the template. The fields are shown in the order in which they appear in the template.

Table 100: Version 9 IPv6 Template Fields for MX, M, and T Series

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 ToS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	139
Input Interface	10

Table 100: Version 9 IPv6 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
VLAN ID	58
IPv6 Source Mask	29
IPv6 Destination Mask	30
Source AS	16
Destination AS	17
IPv6 Next Hop Address	62
IPv6 BGP Next Hop Address	63
TCP Flags	6
Output Interface	14
Minimum TTL	52
Maximum TTL	53
Flow End Reason	136
Flow Direction	61
802.1Q VLAN identifier (dot1qVlanId)	243
802.1Q Customer VLAN identifier (dot1qCustomerVlanId)	245

Table 100: Version 9 IPv6 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
IP Identifier	54
IPv6 Option Headers	64
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21

Fields Included in the Version 9 IPv6 Template for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series

Table 101 on page 512 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 101: IPv6 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3SIB), PTX10016 Series

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 TOS	5
IPv6 Protocol	4

Table 101: IPv6 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3SIB), PTX10016 Series *(Continued)*

Field	Element ID
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10
Source AS	16
Destination AS	17
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv6 Next Hop Address	62
IPv6 BGP NextHop Address	63
IPv6 Source Mask	29
IPv6 Destination Mask	30

Table 101: IPv6 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3SIB), PTX10016 Series *(Continued)*

Field	Element ID
TCP Flags	6
IP Protocol Version	60

Fields Included in the Version 9 IPv6 Template for PTX10003-160C, PTX10003-80C, PTX10004, PTX10008 (with the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD routers

Table 102 on page 514 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 102: IPv6 Template Fields for PTX10003-160C, PTX10003-80C, PTX10004, PTX10008 (with the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD routers

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 TOS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32

Table 102: IPv6 Template Fields for PTX10003-160C, PTX10003-80C, PTX10004, PTX10008 (with the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD routers *(Continued)*

Field	Element ID
Input SNMP Index	10
Output SNMP Index	14
IPv6 Source Mask	29
IPv6 DestinationMask	30
Source AS	16
Destination AS	17
IPv6 Next Hop Address	62
IPv6 BGP Next Hop Address	63
TCP Flags	6
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IP Protocol Version	60
Policy Map Name	32765 (PTX10002-36QDD only)

Table 102: IPv6 Template Fields for PTX10003-160C, PTX10003-80C, PTX10004, PTX10008 (with the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD routers *(Continued)*

Field	Element ID
Minimum TTL	52
Maximum TTL	53

Fields Included in the Version 9 IPv6 Template for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100

Table 103 on page 516 shows the fields that are available in the template. The fields are shown in the order in which they appear in the template.

Table 103: Version 9 IPv6 Template Fields for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100 Routers

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 TOS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code (IPv6)	139
Input Interface	10

Table 103: Version 9 IPv6 Template Fields for ACX7024X, ACX7332, ACX7348, ACX7509, and ACX7100 Routers *(Continued)*

Field	Element ID
IPv6 Source Mask	29
IPv6 Destination Mask	30
TCP Flags	6
IP Protocol Version	60
Source AS	16
Destination AS	17
IPv6 BGP Next Hop Address	63
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv6 Next Hop Address	62

Fields Included in the Version 9 MPLS-IPv4 Template for MX, M, and T Series

Starting in Junos OS Release 16.1, the version 9 MPLS-IPv4 template is supported. Table 104 on page 518 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 104: Version 9 MPLS-IPv4 Template Fields for MX, M, and T Series

Field	Element ID
MPLS Label 1	70
MPLS Label 2	71
MPLS Label 3	72
MPLS Top Label IP Address	47
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 ToS	5
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10
VLAN ID	58
IPv4 Source Mask	9
IPv4 Destination Mask	13

Table 104: Version 9 MPLS-IPv4 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
Source AS	16
Destination AS	17
IPv4 Next Hop Address	15
TCP Flags	6
Output Interface	14
Minimum TTL	52
Maximum TTL	53
Flow End Reason	136
IP Protocol Version	60
BGP IPv4 Next Hop Address	18
Flow Direction	61
802.1Q VLAN identifier (dot1qVlanId)	243
802.1Q Customer VLAN identifier (dot1qCustomerVlanId)	245
IP Identifier	54
Number of Flow Bytes	1

Table 104: Version 9 MPLS-IPv4 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21

Fields Included in the Version 9 MPLS-IPv4 Template for PTX Series and the QFX10002-60C Switch

Starting in Junos OS Release 18.2R1, the version 9 MPLS-IPv4 template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the version 9 MPLS-IPv4 template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the version 9 MPLS-IPv4 template is supported for the QFX10002-60C switch. Table 105 on page 520 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 105: Version 9 MPLS-IPv4 Template Fields for PTX Series and the QFX10002-60C Switch

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 ToS	5
IPv4 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32

Table 105: Version 9 MPLS-IPv4 Template Fields for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
Input Interface	10
Source AS	16
Destination AS	17
BGP IPv4 Next Hop Address	18
IPv6 BGP Next Hop Address (PTX10002-36QDD only)	63
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv4 Next Hop Address	15
IPv4 Source Mask	9
IPv4 Destination Mask	13
TCP Flags	6
MPLS Label 1	70

Table 105: Version 9 MPLS-IPv4 Template Fields for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
MPLS Label 2	71
MPLS Label 3	72
MPLS Top Label IP Address	47
Minimum TTL	52
Maximum TTL	53

Fields Included in the Version 9 MPLS-IPv6 Template for MX, M, and T Series

Starting in Junos OS Release 18.4R1, the version 9 MPLS-IPv6 template is supported for the MX Series. Table 106 on page 522 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 106: Version 9 MPLS-IPv6 Template Fields for MX, M, and T Series

Field	Element ID
MPLS Label 1	70
MPLS Label 2	71
MPLS Label 3	72
MPLS Top Label IP Address (Only IPv4 top label addresses are exported. IPv6 top label addresses report a value of zero.)	47
IPv6 Source Address	27

Table 106: Version 9 MPLS-IPv6 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
IPv6 Destination Address	28
IPv6 ToS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code (IPv6)	139
Input Interface	10
VLAN ID	58
IPv6 Source Mask	29
IPv6 Destination Mask	30
Source AS	16
Destination AS	17
IPv6 Next Hop Address	62
IPv6 BGP Next Hop Address	63
TCP Flags	6

Table 106: Version 9 MPLS-IPv6 Template Fields for MX, M, and T Series *(Continued)*

Field	Element ID
Output Interface	14
Minimum TTL	52
Maximum TTL	53
Flow End Reason	136
Flow Direction	61
802.1Q VLAN identifier (dot1qVlanId)	243
802.1Q Customer VLAN identifier (dot1qCustomerVlanId)	245
IP Identifier	54
IPv6 Option Headers	64
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21

Fields Included in the Version 9 MPLS-IPv6 Template for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series

Starting in Junos OS Release 18.2R1, the version 9 MPLS-IPv6 template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the version 9 MPLS-IPv6 template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the version 9 MPLS-IPv6 template is supported for the QFX10002-60C switch. Table 107 on page 525 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 107: Version 9 MPLS-IPv6 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 ToS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code (IPv6)	32
Input Interface	10
Source AS	16
Destination AS	17
Output Interface	14

Table 107: Version 9 MPLS-IPv6 Template Fields for PTX3000, PTX5000, PTX1000, PTX10001-20C, PTX10002-60C, QFX10002-60C, PTX10008 (without the JNP10008-SF3), PTX10016 Series *(Continued)*

Field	Element ID
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv6 Source Mask	29
IPv6 Destination Mask	30
IPv6 Next Hop Address	62
IPv6 BGP NextHop Address	63
TCP Flags	6
IP protocol version of IP payload on MPLS VPN	60
RSVP label (top MPLS label stack entry) for MPLS tunnel	70
RSVP label pushed before top label	71
MPLS Label 3	72
MPLS Top Label IP Address	47

Fields Included in the Version 9 MPLS-IPv6 Template for PTX10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-LC1202 line card and the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD Routers

Table 108 on page 527 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 108: Version 9 MPLS-IPv6 Template Fields for PTX 10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-1202-36MR line card and the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD Routers

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 ToS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code (IPv6)	32
Input Interface	10
Source AS	16
Destination AS	17
Output Interface	14
Number of Flow Bytes	1

Table 108: Version 9 MPLS-IPv6 Template Fields for PTX 10003, PTX10004, PTX10008 (with the JNP10K-LC1201 or JNP10K-1202-36MR line card and the JNP10008-SF3), PTX10001-36MR, and PTX10002-36QDD Routers *(Continued)*

Field	Element ID
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv6 Source Mask	29
IPv6 Destination Mask	30
IPv6 Next Hop Address	62
IPv6 BGP Next Hop Address	63
TCP Flags	6
IP protocol version of IP payload on MPLS VPN	60
RSVP label (top MPLS label stack entry) for MPLS tunnel	70
RSVP label pushed before top label	71
MPLS Label 3	72
MPLS Top Label IP Address	47
Minimum TTL	52
Maximum TTL	53

Fields Included in the Version 9 MPLS Template for MX, M, and T Series

Starting in Junos OS Release 16.1, the version 9 MPLS template is supported. Table 109 on page 529 shows the fields that are included in the template. The fields are shown in the order in which they appear in the template.

Table 109: Version 9 MPLS Template Fields for MX, M, and T Series

Field	Element ID
MPLS Label 1	70
MPLS Label 2	71
MPLS Label 3	72
Input Interface	10
Output Interface	14
Flow End Reason	136
Number of Flow Bytes	1
Number of Flow Packets	2
First Switched	ww
Time the flow ended with respect to system up time (FPC up time)	21

Fields Included in the Version 9 MPLS Template for PTX Series and the QFX10002-60C Switch

Starting in Junos OS Release 18.2R1, the version 9 MPLS template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS template is supported for the QFX10002-60C switch. Table 110 on page 530 shows the fields that are included in the template.

Table 110: Version 9 MPLS Template Fields for PTX Series and the QFX10002-60C Switch

Field	Element ID
Input Interface	10
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
MPLS Label 1	70
MPLS Label 2	71
MPLS Label 3	72

Fields Included in the Version 9 MPLS-over-UDP Template for PTX Series and the QFX10002-60C Switch for Flows Within an IP Network Having an IPv4 Payload

Starting in Junos OS Release 18.1R1, the version 9 MPLS-over-UDP template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS-over-UDP template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS-over-UDP template is supported for the QFX10002-60C switch.

Inline active flow monitoring for MPLS-over-UDP traffic is not supported on the PTX10001-36MR, PTX10002-36QDD, PTX10003, PTX10004, and PTX10008 (with the JNP10008-SF3) routers.

Table 111 on page 531 shows the fields that are available in the Version 9 template for MPLS-over-UDP flows that are within an IP network and have an IPv4 payload. The fields are shown in the order in which they appear in the template.

Table 111: Version 9 MPLS-over-UDP Carried on IP Network Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch

Field	Element ID
IPv4 source address for tunnel endpoint	8
IPv4 destination address for tunnel endpoint	12
UDP source port for tunnel endpoint	7
Tunnel endpoint destination transport port	11
IPv4 source mask for tunnel source IP address	9
IPv4 destination mask for tunnel destination IP address	13
Source AS for tunnel	16
Destination AS for tunnel	17
IPv4 next hop address—gateway for tunnel destination IP address	15
BGP IPv4 next hop address—tunnel destination IP BGP peer	18
Input SNMP index	10
Output SNMP index	14
MPLS label 1—VPN bottom of stack label	70
IP protocol version of IP payload on MPLS VPN	60
IPv4 source address of tunnel payload	8

Table 111: Version 9 MPLS-over-UDP Carried on IP Network Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
IPv4 destination address of tunnel payload	12
IP protocol of tunnel payload	4
IP TOS	5
Source transport port	7
Destination transport port	11
ICMP type	32
TCP flags	6
Number of flow bytes	1
Number of flow packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21

Fields Included in the Version 9 MPLS-over-UDP Template for PTX Series and the QFX10002-60C Switch for Flows Encapsulated in an RSVP-TE LSP and Having an IPv4 Payload

Starting in Junos OS Release 18.1R1, the version 9 MPLS-over-UDP template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS-over-UDP template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS-over-UDP template is supported for the QFX10002-60C switch.

Inline active flow monitoring for MPLS-over-UDP traffic is not supported on the PTX10001-36MR, PTX10002-36QDD, PTX10003, PTX10004, and PTX10008 (with the JNP10008-SF3) routers.

Table 112 on page 533 shows the fields that are available in the Version 9 template for MPLS-over-UDP flows that are encapsulated in an RSVP-TE LSP in the inner MPLS network and have an IPv4 payload. The fields are shown in the order in which they appear in the template.

Table 112: Version 9 MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch

Field	Element ID
RSVP label (top MPLS label stack entry) for MPLS tunnel	70
RSVP label pushed before top label	71
IPv4 source address for tunnel endpoint	8
IPv4 destination address for tunnel endpoint	12
UDP source port for tunnel endpoint	7
Tunnel endpoint destination transport port	11
IPv4 source mask for tunnel source IP address	9
IPv4 destination mask for tunnel destination IP address	13
Source AS for tunnel	16
Destination AS for tunnel	17
IPv4 next hop address—gateway for tunnel destination IP address	15
BGP IPv4 next hop address—tunnel destination IP BGP peer	18

Table 112: Version 9 MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch ***(Continued)***

Field	Element ID
Input SNMP index	10
Output SNMP index	14
MPLS label 1—VPN bottom of stack label	70
IP protocol version of IP payload on MPLS VPN	60
IPv4 source address of tunnel payload	8
IPv4 destination address of tunnel payload	12
IP protocol of tunnel payload	4
IP TOS	5
Source transport port	7
Destination transport port	11
ICMP type	32
TCP flags	6
Number of flow bytes	1
Number of flow packets	2
Time the flow started with respect to system up time (FPC up time)	22

Table 112: Version 9 MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv4 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
Time the flow ended with respect to system up time (FPC up time)	21

Fields Included in the Version 9 MPLS-over-UDP Template for PTX Series and the QFX10002-60C Switch for Flows Within an IP Network Having an IPv6 Payload

Starting in Junos OS Release 18.1R1, the version 9 MPLS-over-UDP template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS-over-UDP template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS-over-UDP template is supported for the QFX10002-60C switch.

Inline active flow monitoring for MPLS-over-UDP traffic is not supported on the PTX10001-36MR, PTX10002-36QDD, PTX10003, PTX10004, and PTX10008 (with the JNP10008-SF3) routers.

Table 113 on page 535 shows the fields that are available in the Version 9 template for MPLS-over-UDP flows that are within an IP network and have an IPv6 payload. The fields are shown in the order in which they appear in the template.

Table 113: Version 9 MPLS-over-UDP Carried on IP Network Template Fields (IPv6 Payload) for PTX Series and the QFX10002-60C Switch

Field	Element ID
IPv4 source address for tunnel endpoint	8
IPv4 destination address for tunnel endpoint	12
UDP source port for tunnel endpoint	7
Tunnel endpoint destination transport port	11
IPv4 source mask for tunnel source IP address	9
IPv4 destination mask for tunnel destination IP address	13

Table 113: Version 9 MPLS-over-UDP Carried on IP Network Template Fields (IPv6 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
Source AS for tunnel	16
Destination AS for tunnel	17
IPv4 next hop address—gateway for tunnel destination IP address	15
BGP next hop address—tunnel destination IP BGP peer	18
Input SNMP index	10
Output SNMP index	14
MPLS label 1—VPN bottom of stack label	70
IP protocol version of IP payload on MPLS VPN	60
IPv6 source address of tunnel payload	27
IPv6 destination address of tunnel payload	28
IP protocol of tunnel payload	4
IP TOS	5
Source transport port	7
Destination transport port	11
ICMP type V6	32

Table 113: Version 9 MPLS-over-UDP Carried on IP Network Template Fields (IPv6 Payload) for PTX Series and the QFX10002-60C Switch *(Continued)*

Field	Element ID
TCP flags	6
Number of flow bytes	1
Number of flow packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21

Fields Included in the Version 9 MPLS-over-UDP Template for PTX Series and the QFX10002-60C Switch for Flows Encapsulated in an RSVP-TE LSP and Having an IPv6 Payload

Starting in Junos OS Release 18.1R1, the version 9 MPLS-over-UDP template is supported for the PTX Series. Starting in Junos OS Release 19.4R1, the IPFIX MPLS-over-UDP template is supported for the PTX10002-60C router. Starting in Junos OS Release 21.2R1, the IPFIX MPLS-over-UDP template is supported for the QFX10002-60C switch.

Inline active flow monitoring for MPLS-over-UDP traffic is not supported on the PTX10001-36MR, PTX10002-36QDD, PTX10003, PTX10004, and PTX10008 (with the JNP10008-SF3) routers.

Table 114 on page 537 shows the fields that are available in the Version 9 template for MPLS-over-UDP flows that are encapsulated in an RSVP-TE LSP in the inner MPLS network and have an IPv6 payload. The fields are shown in the order in which they appear in the template.

Table 114: Version 9 MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv6 Payload) for PTX Series and QFX10002-60C Switch

Field	Element ID
RSVP label (top MPLS label stack entry) for MPLS tunnel	70

Table 114: Version 9 MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv6 Payload) for PTX Series and QFX10002-60C Switch *(Continued)*

Field	Element ID
RSVP label pushed before top label	71
IPv4 source address for tunnel endpoint	8
IPv4 destination address for tunnel endpoint	12
UDP source port for tunnel endpoint	7
Tunnel endpoint destination transport port	11
IPv4 source mask for tunnel source IP address	9
IPv4 destination mask for tunnel destination IP address	13
Source AS for tunnel	16
Destination AS for tunnel	17
IPv4 next hop address—gateway for tunnel destination IP address	15
BGP next hop address—tunnel destination IP BGP peer	18
Input SNMP index	10
Output SNMP index	14
MPLS label 1—VPN bottom of stack label	70
IP protocol version of IP payload on MPLS VPN	60

Table 114: Version 9 MPLS-over-UDP Encapsulated in RSVP-TE LSP Template Fields (IPv6 Payload) for PTX Series and QFX10002-60C Switch *(Continued)*

Field	Element ID
IPv6 source address of tunnel payload	27
IPv6 destination address of tunnel payload	28
IP protocol of tunnel payload	4
IP TOS	5
Source transport port	7
Destination transport port	11
ICMP type	32
TCP flags	6
Number of flow bytes	1
Number of flow packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
25.4R1-EVO	For the PTX10002-36QDD router, in the ingress direction of sampling at a SRV6 tunnel's head node, we support the IPv6 BGP Nexthop Address (IE 63) field in the IPv4 and MPLS-IPv4 templates, for both the IPFIX and the version 9 formats.
25.2R1-EVO	For the PTX Series routers, Information Element 52, Minimum TTL, and Information Element 53, Maximum TTL, are now available in the IPv4, IPv6, MPLS-IPv4, and the MPLS-IPv6 templates for the IPFIX and version 9 export formats. Because no flow cache is maintained, the two TTL values are the same.
24.4R1-EVO	For the PTX10002-36QDD router, we support ingress and egress sampling (including on IRB interfaces) of IPv4, IPv6, and MPLS traffic and exporting IPFIX and version 9 records. You can configure either IPv4 or IPv6 collectors for each family within a sampling instance; you cannot specify both for the same family. You can specify up to four collectors for each family. You can export BGP community and AS path information using IP Flow Information Export (IPFIX) information elements 483 through 491, 16, and 17, per RFCs 8549 and 6313. Information Element 63, IPv6 BGP NextHop Address, is available in both the IPv6 template and the MPLS-IPv6 template for the IPFIX and version 9 export formats.
24.4R1-EVO	Starting in Junos OS Evolved Release 24.4R1 for the PTX Series routers, we have added support for reporting an accurate BGP next-hop address for traffic load-balanced over multiple BGP peers in the ingress direction. Prior to this release, we reported the first in a list of BGP next hops. To contain this accurate BGP next-hop address, we use the IPv4 BGP Nexthop Address (IE 18) field in the IPv4 and MPLS-IPv4 templates and the IPv6 BGP Nexthop Address (IE 63) field in the IPv6 and MPLS-IPv6 templates, for both the IPFIX and the version 9 formats. To configure this feature, include the `nexthop-learning enable` statement at the `[edit services flow-monitoring (version-ipfix | version9) template]` hierarchy level. When this feature is enabled, the `fragmentIdentification` (IE 54) field reports a value of 0.
24.4R1-EVO	For the PTX10002-36QDD router, we support a new Juniper-specific enterprise Information Element ID, 32765, in the data record templates `ip4-template` and `ipv6-template`. This new IE ID is 4 bytes long and contains the first 4 characters of the policy-map name. Therefore, the first 4 letters of your policy-map names should be unique. You configure this new IE ID with the `include-policy-map-name` statement at the `[edit services flow-monitoring (version-ipfix|version9)` *`template-name`* `data-record-fields]` hierarchy level. You configure policy maps at the `[edit class-of-service policy-map]` hierarchy level.

24.2R1-EVO	For the PTX Series, starting with Junos OS Evolved Release 24.2R1, we support export of IPFIX or version 9 records of inline active flow monitoring sampled packets to collectors reachable through interfaces belonging to the mgmt_junos VRF instance and through WAN ports belonging to a non-default VRF instance.
24.2R1	Starting in Junos OS Release Release 24.2R1 for the MX240, MX480, MX960, MX2010, and MX2020 routers, we have added support for reporting an accurate BGP next-hop address for traffic load-balanced over multiple BGP peers in the ingress direction. Prior to this release, we reported the first in a list of BGP next hops. To contain this accurate BGP next-hop address, we use the IPv4 BGP Nexthop Address (IE 18) field in the IPv4 and MPLS-IPv4 templates and the IPv6 BGP Nexthop Address (IE 63) field in the IPv6 and MPLS-IPv6 templates, for both the IPFIX and the version 9 formats. To configure this feature, include the `multi-bgp-path` statement at the `[edit services flow-monitoring (version-ipfix | version9) template]` hierarchy level. For IPv6 and MPLS-IPv6 flows, you also need to configure the `ipv6-extended-attrib` statement at the `[edit chassis fpc` *`slot-number`* `inline-services flow-table-size]` hierarchy level. When this feature is enabled, the `fragmentIdentification` (IE 54) field reports a value of 0.
24.2R1	Starting in Junos OS Release 24.2R1 for MX Series routers, we support configuring inline active flow monitoring for a demux0 interface mapped to an underlying AE interface, but for only the mpls-template of family MPLS and for protocol family CCC. (No ip4-template or ipv6-template support.) Existing maximum flow scale on the linecard is supported.
23.4R1-EVO	Starting in Junos OS Evolved Release 23.4R1 for the ACX7024X, ACX7332, and ACX7348 routers, we support ingress and egress sampling of IPv4 and IPv6 traffic on aggregated Ethernet and IRB interfaces, for both the IPFIX and version 9 export formats. You can configure up to four IPv4 collectors for inline active flow monitoring. You can sample traffic mapped to non-default VRFs, but you cannot export IPFIX or V9 records of those sampled packets through a non-default or management VRF.
23.1R1-EVO	Starting in Junos OS Evolved Release 23.1R1, for the PTX10001-36MR, PTX10003, and PTX10004 routers, as well as the PTX10008 and PTX10016 routers (with the JNP10K-LC1201 or the JNP10K-LC1202 line card and the JNP10008-SF3) routers, we support IPv6 addresses for IPFIX and version 9 collectors. You can configure either IPv4 or IPv6 collectors for each family within a sampling instance; you cannot specify both for the same family. You can specify up to four collectors for each family. You specify the destination server address with the `flow-server` *`address`* statement and the source address with the `inline-jflow source-address` *`address`* statement at the `[edit forwarding-options sampling instance` *`name`* `family (inet | inet6 | mpls) output]` hierarchy level.
23.1R1-EVO	Starting in Junos OS Evolved Release 23.1R1 for the ACX7100 and ACX7509 routers, we support ingress and egress sampling of IPv4 and IPv6 traffic on aggregated Ethernet and IRB interfaces, for both the IPFIX and version 9 export formats. You can configure up to four IPv4 collectors for inline active flow monitoring. You can sample traffic mapped to non-default VRFs, but you cannot export IPFIX or V9 records of those sampled packets through a non-default or management VRF.

22.4R1	Starting in Junos OS Release 22.4R1 for the MX240, MX480, MX960, MX10004, MX10008, MX2008, MX2010, and MX2020 routers, we support ingress and egress sampling of IPv4, IPv6, and MPLS traffic on abstracted fabric (af) interfaces between guest network functions (GNFs) in a node slicing scenario, for both the IPFIX and version 9 export formats.
22.2R1-EVO	Starting in Junos OS Evolved 22.2R1 for PTX Series routers, two more information elements have been added to the IPFIX IPv4 and IPv6 templates. These elements carry the packet loss priority (PLP) values and the first two characters of the configured forwarding class name that the sampled packet carries. The collector uses these elements to derive the DiffServ code point (DSCP) bits that the packet would contain while exiting the router.
22.2R1-EVO	Starting in Junos OS Evolved Release 22.2R1 for the PTX10003 router, Information Element 63, IPv6 BGP NextHop Address, is available in both the IPv6 template and the MPLS-IPv6 template for the IPFIX and version 9 export formats.
21.4R1-EVO	Starting in Junos OS Evolved Release 21.4R1 for PTX Series, you can export BGP community and AS path information using IP Flow Information Export (IPFIX) information elements 483 through 491, 16, and 17, per RFCs 8549 and 6313. Content providers can use this information to identify a transit service provider degrading the quality of the service. You configure these elements with the statement `data-record-fields` at the `[edit services flow-monitoring version-ipfix template template-name]` hierarchy level.
21.3R1-EVO	Starting in Junos OS Evolved Release 21.3R1 for the PTX10001-36MR, PTX10004, and PTX10008 routers, Information Element 63, IPv6 BGP NextHop Address, is available in both the IPv6 template and the MPLS-IPv6 template for the IPFIX and version 9 export formats.
21.3R1	Starting with Junos OS Release 21.3R1 for PTX Series routers, no flows are maintained. Every sampled packet is considered to be a flow. When the sampled packet is received, the flow is created and immediately timed out as inactive, and the software exports a record to the collector. Therefore, the number of records sent to the collector is higher than before.
21.2R1-Evo	Starting with Junos OS Evolved 21.2R1, no flows are maintained. Every sampled packet is considered to be a flow. When the sampled packet is received, the flow is created and immediately timed out as inactive, and the software exports a record to the collector. Therefore, the number of records sent to the collector is higher than before.
21.2R1	Starting with Junos OS Release 21.2R1 on the QFX10002-60C switch, you can perform inline active flow monitoring for MPLS-over-UDP flows to look past the tunnel header to sample and report on the inner payload at both the transit and egress nodes of the tunnel. MPLS IPv4 and IPv6 payloads and IPFIX and version 9 templates are supported. Only ingress sampling is supported.
21.1R1-EVO	Starting with Junos OS Evolved 21.1R1, for the PTX10004 router, you can configure up to four collectors for inline active flow monitoring.

20.4R1-EVO	Starting with Junos OS Evolved 20.4R1, for the PTX10001-36MR and the PTX10008 (with the JNP10K-LC1202 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring.
20.3R1-EVO	Starting with Junos OS Evolved 20.3R1, for the PTX10003 and PTX10008 (with the JNP10K-LC1201 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring.
20.3R1	Starting in Junos OS Release 20.3R1 for PTX1000, PTX10008 (without the JNP10008-SF3), and PTX10016), two more information elements have been added to the IPFIX IPv4 and IPv6 templates. These elements carry the packet loss priority (PLP) values and the first two characters of the configured forwarding class name that the sampled packet carries. The collector uses these elements to derive the DiffServ code point (DSCP) bits that the packet would contain while exiting the router. To use these elements, you must configure the `next-hop-learning enable` statement at the [edit services flow-monitoring version-ipfix template name] hierarchy level.
19.4R1	Starting with Junos OS Release 19.4R1 on the PTX10002-60C router, you can perform flow monitoring for MPLS-over-UDP flows to look past the tunnel header to sample and report on the inner payload at both the transit and egress nodes of the tunnel. MPLS IPv4 and IPv6 payloads and IPFIX and version 9 templates are supported. Only ingress sampling is supported.
19.3R2	Inline active flow monitoring is not supported when you enable Next Gen Services on an MX Series router.
19.2R1	Starting in Junos OS Release 19.2R1 for MX and PTX Series routers, Information Element 63, IPv6 BGP NextHop Address, is available in both the IPv6 template and the MPLS-IPv6 template for the IPFIX and version 9 export formats.
18.4R1	Starting in Junos OS Release 18.4R1, you can configure inline active flow monitoring for MPLS-IPv6 traffic for MX Series routers.
18.4R1	Starting in Junos OS Release 18.4R1, the IPFIX and version 9 MPLS-IPv6 templates are supported for the MX Series.
18.2R1	Starting in Junos OS Release 18.2R1, you can configure inline active flow monitoring for MPLS, MPLS-IPv4, and MPLS-IPv6 traffic for PTX3000 and PTX5000 Series routers.
18.2R1	Starting in Junos OS Release 18.2R1, you can configure inline active flow monitoring for bridge traffic for MX Series routers.
18.2R1	Starting in Junos OS Release 18.2R1, the IPFIX and version 9 MPLS templates are supported for the PTX Series.

18.2R1	Starting in Junos OS Release 18.2R1, the IPFIX and version 9 MPLS-IPv4 templates are supported for the PTX Series.
18.2R1	Starting in Junos OS Release 18.2R1, the IPFIX and version 9 MPLS-IPv6 templates are supported for the PTX Series.
18.1R1	Starting in Junos OS Release 18.1R1, you can also apply version 9 flow templates to IPv6 traffic.
18.1R1	Starting in Junos OS Release 18.1R1, you can configure inline active flow monitoring for MPLS-over-UDP traffic for PTX3000 and PTX5000 Series routers.
18.1R1	Starting in Junos OS Release 18.1R1, the IPFIX and version 9 MPLS-over-UDP templates are supported for the PTX Series.
17.4R1	Starting with Junos OS Release 17.4R1, you can apply IPFIX flow templates to unicast IPv4 and IPv6 traffic on QFX10008 and QFX10016 switches.
17.2R1	Starting with Junos OS Release 17.2R1, you can apply IPFIX flow templates to unicast IPv4 and IPv6 traffic on QFX10002 switches.
16.1R4	In Junos OS Release 16.2 and in Junos OS Release 16.1R3 and earlier, you can configure only one collector under a family for inline active flow monitoring. Starting with Junos OS Release 16.1R4 and 17.2R1, you can configure up to four collectors under a family for inline active flow monitoring.
16.1R1	Starting in Junos OS Release 16.1R1 and 15.1F2, the IPv4 table is assigned a default value of 1024.
16.1R1	Starting in Junos OS Release 16.1R1, you can also apply IPFIX and version 9 flow templates to MPLS and MPLS-IPv4 traffic.
16.1R1	Flow Direction (Starting in Junos OS Release 16.1R1)
16.1R1	Starting in Junos OS Release 16.1R1, the IPFIX VPLS template is supported.
15.1F7	Starting in Junos OS Release 15.1F7 for the MX5, MX10, MX40, MX80, MX104, MX240, MX480, MX960, MX2010, and MX2020 routers, you can configure the `report-zero-oif-gw-on-discard` at the `[edit chassis fpc` *`slot-number`* `inline-services]` hierarchy level to set the flow record value to 0 (zero) for the IPFIX information elements `ipNextHopIPv6Address` (Element ID 62), `egressInterface` (Element ID 14), and `ipNextHopIPv4Address` (Element ID 15) for only the flow records for rejected, discarded, and policed flows. You configure this statement to help you track what traffic is being dropped.

RELATED DOCUMENTATION

Example: Configuring Inline Active Flow Monitoring on MX Series and T4000 Routers | 585
Configuring Inline Active Flow Monitoring on MX80 and MX104 Routers | 555

Configuring Inline Active Flow Monitoring Using Routers, Switches or NFX250

Inline active flow monitoring is implemented on the Packet Forwarding Engine. The Packet Forwarding Engine performs functions such as creating and updating flows, and updating flow records. The flow records are sent out in industry-standard IPFIX or version 9 format.

On routers with MS-PICs or MS-DPCs, IPv4 and IPv6 fragments are processed accurately. The flow monitoring application creates two flows for every fragmented flow. The first fragment that has the complete Layer 4 information forms the first flow with 5-tuple data and subsequently, all the fragmented packets related to this flow form another flow with the Layer 4 fields set to zero.

The following limitations and restrictions apply to the inline active flow monitoring feature:

- Configuring both sFlow and inline active flow monitoring on the same interface leads to unexpected behavior. Therefore, configure these features on separate interfaces.
- Configuring both egress port mirroring and inline active flow monitoring on the same interface leads to unexpected behavior. Therefore, configure these features on separate interfaces.
- Ingress and egress sampling are sent to the same host-path queue. The packet rate in the queue is shared across ingress and egress sampled packets.
- Forwarding class configuration is not effective. Export record packets are always considered to be control frames and as such are pushed to the network-control queue.
- If multiple inline active flow monitoring firewall filters match to a flow, only the actions of the first filter are taken.
- In ingress sampling, if the destination port is on an aggregated Ethernet interface, the output interface is invalid.

The following considerations apply to the inline active flow monitoring instance configuration:

- Sampling run-length and clip-size are not supported.
- For inline configurations, collectors are not reachable via management interfaces, such as `fxp0`.
- Inline active flow monitoring does not support `cflowd`. Therefore, inline flow monitoring does not support the local dump option, which is available only with cflowd.

- Inline active flow monitoring is not supported when you enable Next Gen Services on an MX Series router.
- The number of collectors that are supported depends on the device:
  - In Junos OS Release 16.2 and in Junos OS Release 16.1R3 and earlier, you can configure only one collector under a family for inline active flow monitoring. Starting with Junos OS Release 16.1R4 and 17.2R1, you can configure up to four collectors under a family for inline active flow monitoring. Starting with Junos OS Evolved 20.3R1, for the PTX10003 and PTX10008 (with the JNP10K-LC1201 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring. Starting with Junos OS Evolved 20.4R1, for the PTX10001-36MR and the PTX10008 (with the JNP10K-LC1202 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring. Starting with Junos OS Evolved 21.1R1, for the PTX10004 router, you can configure up to four collectors for inline active flow monitoring. The Packet Forwarding Engine (PFE) can export the flow record, flow record template, option data, and option data template packet to all configured collectors. To configure a collector under a family for inline active flow monitoring, configure the `flow-server` statement at the `[edit forwarding-options sampling instance *instance-name* family (inet | inet6 | mpls) output]` hierarchy level. To specify up to four collectors, include up to four `flow-server` statements.
  - For inline configurations on all other devices, each family can support only one collector.

Inline active flow monitoring is configured using statements from four hierarchy levels:

- `[edit chassis]` —At this level, you associate the sampling instance with the FPC on which the media interface is present (except on the MX80 and MX104—see "Configuring Inline Active Flow Monitoring on MX80 and MX104 Routers" on page 555). If you are configuring sampling of IPv4 flows, IPv6 flows or VPLS flows (Junos OS only), you can configure the flow hash table size for each family, as described below.
- `[edit firewall]`—At this level, you configure a firewall filter for the family of traffic to be sampled. You must attach this filter to the interface on which you want to sample the traffic.
- `[edit forwarding-options]`—At this level, you configure a sampling instance and associate the template with the sampling instance. At this level, you also configure the flow-server IP address and port number as well as the flow export rate.
- `[edit services flow-monitoring]` —At this level, you configure the template properties for inline flow monitoring.

Before you configure inline active flow monitoring, you should ensure that you have adequately-sized hash tables for IPv4, IPv6, MPLS, and VPLS flow sampling. (VPLS flow sampling is Junos OS only). These tables can use one to fifteen 256K areas. Starting with Junos OS Release 16.1R1 and 15.1F2, the IPv4 table is assigned a default value of 1024. Prior to Junos OS Release 16.1 and 15.1F2, the IPv4 table is assigned a default value of fifteen 256K areas. The IPv6 table is assigned a default value of 1024, and the VPLS table is assigned a default value of 1024. When anticipated traffic volume requires larger tables, allocate larger tables.

To allocate flow hash tables:

1. Go to the [edit chassis fpc 0 inline-services flow-table-size] hierarchy level for inline services on the FPC that processes the monitored flows.

```
[edit]
user@host# edit chassis fpc 0 inline-services flow-table-size
```

2. Specify the required sizes for the sampling hash tables.

```
[edit  chassis fpc 0 inline-services flow-table-size]
user@host# set bridge-flow-table-size units
user@host# set ipv4-flow-table-size  units
user@host# set ipv6-flow-table-size units
user@host# set mpls-flow-table-size  units
user@host# set vpls-flow-table-size units
```

NOTE: Starting in Junos OS Release 18.2R1, the `bridge-flow-table-size` option is available and the `vpls-flow-table-size` option is deprecated; use the `bridge-flow-table-size` option instead. The `bridge-flow-table-size` option supports both VPLS and bridge records.

NOTE:

Starting with Junos OS Release 17.3R1, the maximum supported flow table size varies by line card type. See *flow-table-size* for how the size varies by line card type.

Also, starting in Junos OS Release 16.1R1 and 15.1F2, changing the flow hash table size does *not* automatically reboot the FPC (for earlier releases changing the flow hash table size triggers the FPC to reboot).

To configure inline active flow monitoring on MX Series routers (except for MX80 and MX104 routers), EX Series switches, and T4000 routers with Type 5 FPC:

1. Enable inline active flow monitoring and specify the source address for the traffic.

```
[edit forwarding-options sampling instance instance-name family (bridge | inet | inet6 |
mpls | vpls ) output]
user@host# set inline-jflow source address address
```

2. Specify the template to use with the sampling instance.

```
[edit forwarding-options sampling instance instance-name family (bridge | inet | inet6 |
mpls | vpls ) output flow-server hostname]
user@host# set (version9 | version-ipfix) template template-name
```

3. Configure a template to specify output properties.

```
[edit services flow-monitoring]
user@host# set (version-ipfix | version9) template template-name
```

4. (Optional) Configure the interval after which an active flow is exported.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-active-timeout seconds
```

5. (Optional) Configure the interval of activity that marks a flow as inactive.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-inactive-timeout seconds
```

6. (Optional) Configure the template refresh rate in either number of packets or number of seconds.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set template-refresh-rate (packets packets | seconds seconds)
```

7. (Optional) Configure the refresh rate in either number of packets or number of seconds.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set option-refresh-rate (packets packets | seconds seconds)
```

8. Specify the type of record that the template is used for.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set (bridge-template | ipv4-template | ipv6-template | mpls-ipv4-template | mpls-template | peer-as-billing-template | vpls-template)
```

The `vpls-template` option is only for IPFIX templates.

Starting in Junos OS Release 18.2R1, the `bridge-template` option is available and the `vpls-template` option is deprecated; use the `bridge-template` option instead. The `bridge-template` option (Junos OS only) supports both VPLS and bridge records and is for both IPFIX and version9 templates.

Starting in Junos OS Release 18.4R1, the `mpls-ipv4-template` option is deprecated for inline flow monitoring. To configure MPLS records starting in Junos OS Release 18.4R1, use the `mpls-template` option and the `tunnel-observation` option. This is described in step "9" on page 549.

9. Starting in Junos OS Release 18.4R1 for the MX Series, if you are configuring any type of MPLS flow records, perform the following:

   a. Specify the MPLS template.

   ```
   [edit services flow-monitoring (version-ipfix | version9) template template-name]
   user@host# set mpls-template
   ```

   b. Configure the type of MPLS flow records to create.

   ```
   [edit services flow-monitoring (version-ipfix | version9) template template-name]
   user@host# set tunnel-observation [ipv4 | ipv6]
   ```

   The `tunnel-observation` values enable the creation of the following types of flow records:

   - `ipv4`—MPLS-IPv4 flows
   - `ipv6`—MPLS-IPv6 flows

   You can configure multiple values for `tunnel-observation`.

For an MPLS traffic type that does *not* match any of the `tunnel-observation` values, plain MPLS flow records are created. For example, if you only configure `ipv4`, then MPLS-IPv6 traffic results in plain MPLS flow records.

If you do not configure `tunnel-observation`, plain MPLS flow records are created.

c. If you are running inline flow monitoring on a Lookup (LU) card, enable sideband mode to create MPLS-IPv6 flow records.

```
[edit chassis fpc slot-number inline-services]
user@host# set use-extended-flow-memory
```

If you are running inline flow monitoring on an LU card and do not enable sideband mode, then MPLS-IPv6 traffic results in plain MPLS flow records.

10. (Optional) Include the flow direction value in the template.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-key flow-direction
```

The reported data field contains 0x00 (ingress) or 0x01 (egress). If you do not include the `flow-key flow-direction` statement, the flow direction data field contains the invalid value 0xFF.

11. (Optional) Include VLAN IDs in both the ingress and egress directions in the flow key.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-key vlan-id
```

This statement is not required for ingress and egress VLAN ID reporting on interfaces.

12. Associate the sampling instance with the FPC on which you want to implement inline active flow monitoring.

For MX240, MX480, MX960, MX2010, MX2020, use the following command:

```
[edit ]
user@host# set chassis fpc fpc-number sampling-instance instance-name
```

a. Confirm the configuration by running the following show command:

```
user@host# show chassis
```

```
fpc 0 {
        sampling-instance sample-ins1;
    }
```

For MX5, MX10, MX40, and MX80, use the following command:

```
[edit ]
user@host# set chassis tfeb slot 0 sampling-instance instance-name
```

a. Confirm the configuration by running the following show command:

```
user@host# show chassis
```

```
tfeb {
    slot 0 {
        sampling-instance sample-ins1;
    }
}
```

For MX104, use the following command:

```
[edit ]
user@host# set chassis afeb slot 0 sampling-instance instance-name
```

a. Confirm the configuration by running the following show command:

```
user@host# show chassis
```

```
afeb {
    slot 0 {
        sampling-instance sample-ins1;
    }
}
```

This example shows the sampling configuration for an instance that supports inline active flow monitoring on `family inet`:

```
[edit]
user@host> show forwarding-options
sampling {
    instance {
        sample-ins1 {
            input {
                rate 1;
            }
            family inet {
                output {
                    flow-server 192.0.2.2 {
                        port 2055;
                        version-ipfix {
                            template {
                                ipv4;
                            }
                        }
                    }
                    inline-jflow {
                        source-address 10.11.12.13;
                    }
                }
            }
        }
    }
}
```

Here is the output format configuration:

```
[edit]
user@host> show services flow-monitoring
services {
    flow-monitoring {
        version-ipfix {
            template ipv4 {
                flow-active-timeout 60;
                flow-inactive-timeout 60;
                ipv4-template;
                template-refresh-rate {
                    packets 1000;
                    seconds 10;
                }
                option-refresh-rate {
                    packets 1000;
                    seconds 10;
                }
            }
        }
    }
}
```

The following example shows the output format configuration for chassis `fpc0`:

```
[edit]
user@host> show services flow-monitoring
sampling-instance instance-1; {
    inline-services {
        flow-table-size {
            ipv4-flow-table-size 8;
            ipv6-flow-table-size 7;
        }
    }
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
23.4R1-EVO	Starting in Junos OS Evolved 23.4R1 for the ACX7024X, ACX7332, and ACX7348 routers, we support ingress and egress sampling of IPv4 and IPv6 traffic on aggregated Ethernet and IRB interfaces, for both the IPFIX and version 9 export formats. You can configure up to four IPv4 collectors for inline active flow monitoring. You can sample traffic mapped to non-default VRFs, but you cannot export IPFIX or V9 records of those sampled packets through a non-default or management VRF.
23.1R1-EVO	Starting in Junos OS Evolved Release 23.1R1, for the PTX10001-36MR, PTX10003, and PTX10004 routers, as well as the PTX10008 and PTX10016 routers (with the JNP10K-LC1201 or the JNP10K-LC1202 line card and the JNP10008-SF3) routers, we support IPv6 addresses for IPFIX and version 9 collectors. You can configure either IPv4 or IPv6 collectors for each family within a sampling instance; you cannot specify both for the same family. You can specify up to four collectors for each family. You specify the destination server address with the `flow-server` *`address`* statement and the source address with the `inline-jflow source-address` *`address`* statement at the `[edit forwarding-options sampling instance` *`name`* `family (inet | inet6 | mpls) output]` hierarchy level.
23.1R1-EVO	Starting in Junos OS Evolved 23.1R1 for the ACX7100 and ACX7509 routers, we support ingress and egress sampling of IPv4 and IPv6 traffic on aggregated Ethernet and IRB interfaces, for both the IPFIX and version 9 export formats. You can configure up to four IPv4 collectors for inline active flow monitoring. You can sample traffic mapped to non-default VRFs, but you cannot export IPFIX or V9 records of those sampled packets through a non-default or management VRF.
21.4R1-EVO	Starting in Junos OS Evolved Release 21.4R1 for PTX Series, you can export BGP community and AS path information using IP Flow Information Export (IPFIX) information elements 483 through 491, 16, and 17, per RFCs 8549 and 6313. Content providers can use this information to identify a transit service provider degrading the quality of the service. You configure these elements with the statement `data-record-fields` at the `[edit services flow-monitoring version-ipfix template template-name]` hierarchy level.
21.1R1-EVO	Starting with Junos OS Evolved 21.1R1, for the PTX10004 router, you can configure up to four collectors for inline active flow monitoring.
20.4R1-EVO	Starting with Junos OS Evolved 20.4R1, for the PTX10001-36MR and the PTX10008 (with the JNP10K-LC1202 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring.
20.3R1-EVO	Starting with Junos OS Evolved 20.3R1, for the PTX10003 and PTX10008 (with the JNP10K-LC1201 line card and the JNP10008-SF3) routers, you can configure up to four collectors for inline active flow monitoring.

19.3R2	Inline active flow monitoring is not supported when you enable Next Gen Services on an MX Series router.
18.4R1	Starting in Junos OS Release 18.4R1, the `mpls-ipv4-template` option is deprecated for inline flow monitoring. To configure MPLS records starting in Junos OS Release 18.4R1, use the `mpls-template` option and the `tunnel-observation` option.
18.2R1	Starting in Junos OS Release 20.3R1 for QFX10002-60C switches, you can configure inline active flow monitoring for IPv4 and IPv6 traffic. Both IPFIX and version 9 templates are supported.
18.2R1	Starting in Junos OS Release 18.2R1, the `bridge-flow-table-size` option is available and the `vpls-flow-table-size` option is deprecated; use the `bridge-flow-table-size` option instead.
18.2R1	Starting in Junos OS Release 18.2R1, the `bridge-template` option is available and the `vpls-template` option is deprecated; use the `bridge-template` option instead.
17.2R1	Starting in Junos OS Release 17.2R1 for QFX10002 switches, we added support for inline active flow monitoring with IPFIX templates.
16.1R4	In Junos OS Release 16.2 and in Junos OS Release 16.1R3 and earlier, you can configure only one collector under a family for inline active flow monitoring. Starting with Junos OS Release 16.1R4 and 17.2R1, you can configure up to four collectors under a family for inline active flow monitoring.
16.1R1	Also, starting in Junos OS Release 16.1R1 and 15.1F2, changing the flow hash table size does *not* automatically reboot the FPC.
16.1R1	Starting with Junos OS Release 16.1R1 and 15.1F2, the IPv4 table is assigned a default value of 1024.

RELATED DOCUMENTATION

Configuring Inline Active Flow Monitoring on MX80 and MX104 Routers | 555
Example: Configuring Inline Active Flow Monitoring on MX Series and T4000 Routers | 585
inline-jflow

Configuring Inline Active Flow Monitoring on MX80 and MX104 Routers

To configure inline active flow monitoring on MX80 and MX104 routers:

1. Associate a sampling instance with the Forwarding Engine Processor.

   For the MX80:

   ```
   [edit]
   user@host# set chassis tfeb slot 0 sampling-instance sampling-instance
   ```

   The Forwarding Engine Processor slot is always `0` because MX80 and MX104 routers have only one Packet Forwarding Engine. In this MX80 configuration, the sampling instance is `sample-ins1`.

   ```
   [edit]
   user@host# set chassis tfeb slot 0 sampling-instance sample-ins1
   ```

   For the MX104:

   ```
   [edit]
   user@host# set chassis afeb slot 0 sampling-instance sampling-instance
   ```

   NOTE: MX80 and MX104 routers support only one sampling instance.

2. Under forwarding-options, configure a sampling instance for the flow server and inline jflow instances (these be configured in the following steps):

   ```
   [edit forwarding-options sampling]
   user@host# edit instance inline_sample
   ```

3. Configure the rate at the `[edit forwarding-options sampling instance instance-name input]` hierarchy level to apply specific values for the sampling instance `sample-ins1`.

   ```
   [edit forwarding-options sampling instance sample-ins1 input]
   user@host# set rate number
   ```

In this configuration, the rate is `1000`.

```
[edit forwarding-options sampling instance sample-ins1 input]
user@host# set rate 1000
```

4. Navigate to the output hierarchy and from there, enable a flow server and then specify the output address and port:

```
[edit] forwarding-options sampling instance inline_sample family inet output]
user@host# edit flow-server hostname
```

```
[edit forwarding-options sampling instance inline_sample family inet output flow-server
hostname]
user@host# set port number
```

5. Return to the output hierarchy and specify the source address for inline jflow:

```
[edit forwarding-options sampling instance sample-ins1  family inet output]
user@host# set inline-jflow source-address address
```

In this configuration, the source address is `10.11.12.13`.

```
[edit forwarding-options sampling instance sample-ins1 family inet output]
user@host# set inline-jflow source-address 10.11.12.13
```

6. Specify the output properties.

```
[edit services flow-monitoring]
user@host# set version-ipfix
```

The output format properties are common to other output formats and are described in “"Configuring Inline Active Flow Monitoring to Use IPFIX Flow Templates on MX, vMX and T Series Routers, EX Series Switches, NFX Series Devices, and SRX Devices" on page 621”.

The following is an example of the sampling configuration for an instance that supports inline active flow monitoring on MX80 routers:

```
[edit forwarding-options]
user@host# show
sampling {
    instance   {
        inline_sample {
            input {
                rate 1000;
            }
            family inet{
                output {
                    flow-server 192.168.64.143 {
                        port 80;
                    }
                    inline-jflow  {
                        source-address 10.10.11.12;
                    }
                }
            }
        }
    }
}
```

NOTE: You need not configure a Flexible PIC Concentrator (FPC) slot because MX80 routers have only one Packet Forwarding Engine.

The following considerations apply to the inline flow-monitoring instance configuration:

- This configuration does not support MPLS-IPv6.
- Clip-size is not supported.

RELATED DOCUMENTATION

Configuring Inline Active Flow Monitoring to Use IPFIX Flow Templates on MX, vMX and T Series Routers, EX Series Switches, NFX Series Devices, and SRX Series Firewalls | 621

Configuring Inline Active Flow Monitoring Using Routers, Switches or NFX250 | 74

inline-jflow

Configuring Inline Active Flow Monitoring on ACX and PTX Series Routers

SUMMARY

This topic describes how to configure inline active flow monitoring on ACX and PTX Series routers that support this feature for IPv4 and IPv6 traffic.

IN THIS SECTION

Overview

IN THIS SECTION

Use Feature Explorer: JFlow and Feature Explorer: Traffic Sampling to confirm platform and release support for specific features. See *flow-monitoring* for the configuration statement syntax, because the supported statements vary by platform.

VRF support: Starting in Junos OS Evolved 24.2R1, we support export of IPFIX or version 9 records of inline active flow monitoring sampled packets to collectors reachable through:

- Interfaces belonging to the mgmt_junos VRF instance.
- WAN ports belonging to the non-default VRF instance.

You configure this feature using the `routing-instance` configuration statement at the `[edit forwarding-options sampling instance` *`name`* `family` *`type`* `output flow-server` *`IP-address`*`]` hierarchy level. You must ensure that the collectors are reachable through the management interface. We support a maximum of four collectors for each type of VRF instance. You can configure collectors for both types of VRF instances in the same sampling configuration. However, the collectors reachable through the mgmt_junos VRF instance and the collectors reachable through the WAN ports cannot coexist under the same family, because you can specify only one source IP address per family. You can specify `inet` collectors, `inet6` collectors, or a mix of the two types. Ingress and egress sampling in both the default and non-default VRFs are supported on physical, AE, and IRB interfaces. To configure the mgmt_junos VRF instance, see *Management Interface in a Dedicated Instance*.

MPLS-over-UDP support: To configure inline flow monitoring for MPLS-over UDP traffic on PTX Series Routers, see "Inline Active Flow Monitoring of MPLS-over-UDP Flows on PTX Series Routers" on page 568. Inline active flow monitoring for MPLS-over-UDP traffic is not supported on the PTX10001-36MR, PTX10003, PTX10004, and the PTX10008 (with the JNP10008-SF3) routers.

Collectors: Starting in Junos OS Release 18.2R1, you can configure up to four collectors under a family for inline active flow monitoring. In previous releases of Junos OS, you could configure only one collector under a family for inline active flow monitoring. Starting in Junos OS Evolved 20.3R1, you can configure up to four collectors for inline active flow monitoring. To configure a collector under a family for inline active flow monitoring, configure the `flow-server` statement at the `edit forwarding-options sampling-instance` *`instance-name`* `family (inet | inet6) output` hierarchy level. To specify up to four collectors, include up to four `flow-server` statements.

Line Card CPU: Inline active flow monitoring is implemented on the Line Card CPU (LCPU), except for ACX Series routers running Junos OS Evolved, where all functions are based in software, not hardware. For all other supported platforms, all the functions like flow creation, flow update, and flow records export are done by the LCPU. The flow records are sent out in either the IPFIX format or the version 9 format.

Flows: Starting with Junos OS Evolved Release 21.2R1 and Junos OS Release 21.3R1, no flows are maintained. Every sampled packet is considered to be a flow. When the sampled packet is received, the flow is created and immediately timed out as inactive, and the software exports a record to the collector. Therefore, the number of records sent to the collector is higher than before. The IPFIX and version 9 Options Template Data Record now contains 0 in the `Flow Active Timeout` (Element ID 36) and `Flow Inactive Timeout` (Element ID 37) fields. Therefore, the Options Template Data Record is not compliant with IPFIX RFC 7011. The `show services accounting flow inline-jflow fpc-slot` *`slot`* operational mode command now displays 0 for all of the `Active Flows` and `Timed Out` fields. The values of the various `Total Flows` fields are now equal to their respective `Flow Packets` field values. The values of the various `Flows Inactive Timed Out` fields are now equal to their respective `Flow Packets` field values. The effect of the `nexthop-learning` statement at the `[edit services flow-monitoring version` *`version`* `template` *`template-name`*`]` hierarchy level on this no-flow behavior varies depending upon the operating system. For Junos OS Evolved, we do not recommend that you configure the `nexthop-learning` statement, as it reduces the number of packets that can be processed. For Junos OS, you can configure the `nexthop-learning` statement to change this default no-flow behavior and once again create and maintain flows, then attach the template to all sampling instances associated with FPCs that require the previous behavior.

Limitations and Restrictions

The following limitations and restrictions apply to the inline active flow monitoring feature in Junos OS and Junos OS Evolved:

- Egress MPLS filters are not supported on the PTX10001-36MR, PTX10003, PTX10004, and the PTX10008 (with the JNP10008-SF3) routers.

- The PTX10001-36MR router does not support multiple FPC sampling collection because it has only 1 Routing Engine.
- True outgoing interface (OIF) reporting is not supported for egress sampling. In Junos OS Evolved, true outgoing interface (OIF) reporting is not supported for GRE de-encapsulated packets.
- The interface type field for the true incoming interface is not part of the version 9 template because this element is not present in the version 9 export version.
- For GRE tunnel traffic on PTX10003 routers, the physical interface is reported in the layer 2 header and is considered as one of the keys during flow creation. Therefore, when physical interfaces are moved in or out of the aggregated Ethernet bundle, a new flow is created and the old flows are timed out after a period of inactivity. Physical interface, logical interface, or the aggregated logical interface (based on the configuration) is reported as the incoming interface in export records based on the configuration.

  For GRE tunnel traffic on PTX10008 (with the JNP10008-SF3) routers, an FTI interface is configured to terminate a GRE tunnel. This interface is used during flow creation as one of the keys instead of the physical interface. Hence when a physical interface is moved in or out of an aggregated Ethernet bundle, no new flow is created as the key remains unchanged. Physical interface, logical interface, or the aggregated logical interface (based on the configuration) is reported as the incoming interface in exported records.
- ACX Series routers do not support MPLS templates.
- For ACX routers, you cannot configure sampling directly on a logical interface. Instead, you must configure a firewall filter on the interface to enable sampling.

How to Configure Inline Active Flow Monitoring on ACX or PTX Series Routers

SUMMARY

In this example, we configure a `version-ipfix` template for recording IPv4 and IPv6 traffic flows.

IN THIS SECTION

Configure a Template to Specify Output Properties

1. Define the template and configure the type of flow the template should record. (ACX Series routers do not support MPLS templates.)

```
[edit services flow-monitoring]
user@host# set version-ipfix template template-name ipv4-template
user@host# set version-ipfix template template-name ipv6-template
user@host# set version-ipfix template template-name mpls-template
```

2. (Optional, only for PTX Series routers that support these features) Configure additional output properties for the template, such as flow timeout interval and template/option refresh rates, to control the flow records.

   You can use the `template-refresh-rate` option to configure the frequency at which the flow generator sends updates about template definitions to the flow collector either using number of packets or seconds.

```
[edit services flow-monitoring]
user@host# set version-ipfix template template-name flow-active-timeout seconds
user@host# set version-ipfix template template-name flow-inactive-timeout seconds
user@host# set version-ipfix template template-name template-refresh-rate (packets packets |
seconds seconds)
user@host# set version-ipfix template template-name option-refresh-rate (packets packets |
seconds seconds)
```

3. (Optional, only for PTX Series routers that support these features)
   If you are monitoring MPLS flows, that is, if the template in use is configured for the MPLS protocol family, use the `tunnel-observation` option to identify the types of MPLS flows.

```
[edit services flow-monitoring]
user@host# set version-ipfix template template-name tunnel-observation (ipv4 | ipv6 | mpls-over-
udp)
```

4. (Optional) Enable the learning of next-hop addresses so that the true outgoing interface is reported.

> **NOTE**: Starting in Junos OS Evolved 21.2R1, we do not recommend that you enable learning of next-hop addresses, as it reduces the number of packets that can be processed. However, starting in Junos OS Release 21.3R1, you can configure the `nexthop-learning` statement to change the default no-flow behavior and once again create and maintain flows, then attach the template to all sampling instances associated with FPCs that require the previous behavior.

```
[edit services flow-monitoring]
user@host# set version-ipfix template template-name nexthop-learning enable
```

Configure a Sampling Instance to Specify Input Properties

1. Define the sampling instance and configure the ratio of number of packets to be sampled. For example, if you specify a rate of 10, every tenth packet (1 packet out of 10) is sampled.

```
[edit forwarding-options sampling]
user@host# set instance instance-name input rate number
```

> **BEST PRACTICE**: We recommend that you use a value of 1000 or higher for MPLS flows.

2. Configure the protocol family for the sampling instance and specify a flow collector to send the traffic aggregates. (ACX Series routers support only `family inet` or `family inet6` configuration.)

```
[edit forwarding-options sampling]
user@host# set instance instance-name family (inet | inet6 | mpls) output flow-server hostname
```

3. (Optional) Specify the UDP port for the flow collector and the template to use with the sampling instance. (ACX Series routers do not support MPLS templates.)

```
[edit forwarding-options sampling]
user@host# set instance instance-name family (inet | inet6 | mpls) output flow-server
hostname port port-number
```

```
user@host# set instance instance-name family (inet | inet6 | mpls) output flow-server
hostname version-ipfix template template-name
```

4. Configure inline processing of the sampled packets.

   (ACX Series routers support only `family inet` or `family inet6` configuration.)

```
[edit forwarding-options sampling]
user@host# set instance instance-name family (inet | inet6 | mpls) output inline-jflow source-
address address
```

Assign the Sampling Instance to an FPC

1. Assign the sampling instance to the FPC on which you want to implement flow monitoring.

```
[edit chassis]
user@host# set fpc slot-number sampling-instance instance-name
```

Configure a Firewall Filter to Accept and Sample Flows

1. Configure the firewall filter for the protocol family and enable sampling of traffic flows. (ACX Series routers support only `family inet` or `family inet6` configuration for the inline active flow monitoring feature.)

```
[edit firewall]
user@host# set family (inet | inet6 | mpls) filter filter-name
user@host# set family (inet | inet6 | mpls) filter filter-name term term-name then accept
user@host# set family (inet | inet6 | mpls) filter filter-name term term-name then sample
```

Assign the Firewall Filter to an Interface

1. Assign the input firewall filter to the interface you want to monitor. (ACX Series routers support only `family inet` or `family inet6` configuration for the inline active flow monitoring feature.)

```
[edit interfaces]
user@host# set interface-name unit unit-number family (inet |inet6 | mpls) filter input
filter-name
```

Results from a Sample Configuration

The following is an example of the sampling configuration on a PTX Series router for an instance that supports inline active flow monitoring on `family inet` and on `family inet6`:

```
[edit chassis]
fpc 0 {
    sampling-instance sample-1;
}
```

```
[edit services]
flow-monitoring {
    version-ipfix {
        template test-template {
            flow-active-timeout 30;
            flow-inactive-timeout 60;
            nexthop-learning {
                enable;
            }
            template-refresh-rate {
                seconds 10;
            }
            ipv4-template;
        }
        template v6 {
            ipv6-template;
        }
    }
}
```

```
[edit interfaces]
et-1/0/0 {
    unit 0 {
        family inet {
            filter {
                input ipv4-filter;
                output ipv4-filter;
            }
            address 192.168.100.10/24;
```

```
        }
    }
}
et-1/0/2 {
    unit 0 {
        family inet6 {
            filter {
                input ipv6-filter;
                output ipv6-filter;
            }
            address 2001:db8:0:2::1/64;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.100.1/32;
        }
    }
}
```

```
[edit forwarding-options]
sampling {
    instance {
        sample-1 {
            input {
                rate 10;
            }
            family inet {
                output {
                    flow-server 10.208.174.127 {
                        port 2055;
                        version-ipfix {
                            template {
                                test-template;
                            }
                        }
                   }
                    inline-jflow {
                        source-address 192.168.100.1;
```

```
                    }
                }
            }
            family inet6 {
                output {
                    flow-server 10.208.174.127 {
                        port 2055;
                        version-ipfix {
                            template {
                                v6;
                            }
                        }
                    }
                    inline-jflow {
                        source-address 192.168.100.1;
                    }
                }
            }
        }
    }
}
```

```
[edit firewall]
family inet {
   filter ipv4-filter {
      term ipv4-accept {
         then {
            accept;
            sample;
         }
      }
   }
}
family inet6 {
   filter ipv6-filter {
      term ipv6-accept {
         then {
            accept;
            sample;
         }
      }
```

```
    }
}
```

You can use the *show services accounting flow* command to verify active flow statistics.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.2R1-EVO	We support export of IPFIX or version 9 records of inline active flow monitoring sampled packets to collectors reachable through interfaces belonging to the mgmt_junos VRF instance and through WAN ports belonging to a non-default VRF instance.
21.3R1	No flows are maintained. Every sampled packet is considered to be a flow. When the sampled packet is received, the flow is created and immediately timed out as inactive, and the software exports a record to the collector. Therefore, the number of records sent to the collector is higher than before.
21.2R1-EVO	No flows are maintained. Every sampled packet is considered to be a flow. When the sampled packet is received, the flow is created and immediately timed out as inactive, and the software exports a record to the collector. Therefore, the number of records sent to the collector is higher than before.
20.3R1-EVO	You can configure up to four collectors for inline active flow monitoring.
18.2R1	You can configure up to four collectors under a family for inline active flow monitoring. In previous releases of Junos OS, you could configure only one collector under a family for inline active flow monitoring.

Inline Active Flow Monitoring of MPLS-over-UDP Flows on PTX Series Routers

IN THIS SECTION

You can enable inline active flow monitoring that reports the inner payload of MPLS-over-UDP flows on PTX Series routers and QFX10002-60C switches.

MPLS-over-UDP Flow Monitoring Overview

IN THIS SECTION

Starting with Junos OS Release 18.1R1 on PTX Series routers with an FPC3, PTX10K-LC1101, PTX10K-LC1102, or PTX1000 card, you can perform inline active flow monitoring for MPLS-over-UDP flows to look past the tunnel header to sample and report on the inner payload at both the transit and egress nodes of the tunnel.

Starting with Junos OS Release 19.4R1, on the PTX10002-60C, you can perform inline active flow monitoring for MPLS-over-UDP flows to look past the tunnel header to sample and report on the inner payload at both the transit and egress nodes of the tunnel.

Starting with Junos OS Release 19.4R1, the PTX10002-60C supports inline active flow monitoring for MPLS, MPLS-IPv4, MPLS-IPv6, and MPLS-over-UDP traffic. Both IPFIX and version 9 templates are supported.

Starting with Junos OS Release 21.2R1, the QFX10002-60C supports inline active flow monitoring for MPLS, MPLS-IPv4, MPLS-IPv6, and MPLS-over-UDP traffic. Both IPFIX and version 9 templates are supported.

For a description of the fields included in the templates, see "Understand Inline Active Flow Monitoring" on page 439. Only ingress sampling is supported.

MPLS-over-UDP is not supported on the PTX10001-36MR, PTX10003, PTX10004, and PTX10008 (with the JNP10008-SF3) routers.

Benefits of Using MPLS-Over-UDP Flow Monitoring

- Gather and export detailed information on even the original IPv4 or IPv6 payload of the MPLS-over-UDP flow.

Flow Monitoring Scenarios for MPLS-over-UDP

Monitoring for MPLS-over-UDP tunnels includes the following scenarios:

- The MPLS-over-UDP flow is carried through a full IP network, using IPv4 endpoints on PTX Series routers (see Figure 55 on page 570). The inner payload may be IPv4 or IPv6. Figure 56 on page 570 shows the encapsulated packet. Flow monitoring reports the inner IP header and payload, in addition to the tunnel and MPLS fields.

  You can enable ingress monitoring for the MPLS-over-UDP tunnel at its transit and egress nodes. For example, in Figure 55 on page 570, you can enable ingress monitoring on routers R4, R5, R6, and R7.

Figure 55: MPLS-over-UDP in Full IP Network

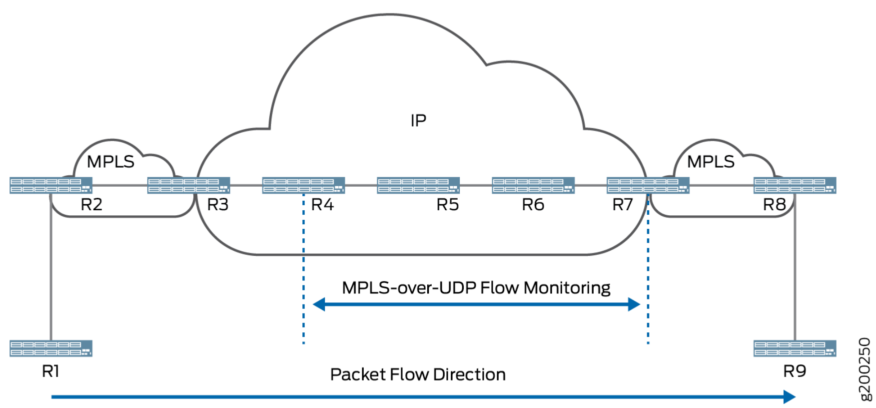
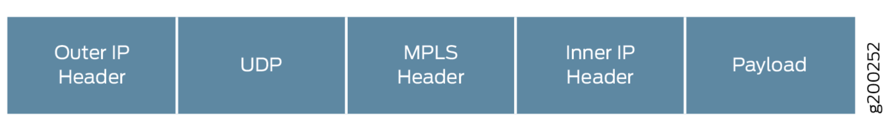


Figure 56: Encapsulated Packet for MPLS-over-UDP in Full IP Network



- The MPLS-over-UDP flow is carried through an IP-MPLS-IP network, using IPv4 endpoints on PTX Series routers (see Figure 57 on page 571). The inner payload may be IPv4 or IPv6. In the inner MPLS network, the MPLS-over-UDP flow is encapsulated in an RSVP-TE label-switched path (LSP). Figure 58 on page 571 shows the encapsulated packet. Flow monitoring reports the inner IP header and payload, in addition to the RSVP label, tunnel, and MPLS fields.

  You can enable ingress monitoring for the MPLS-over-UDP tunnel at its transit and egress nodes. For example, in Figure 57 on page 571, you can enable ingress monitoring on routers R4, R5, R6, R7, R8, and R9.

Figure 57: MPLS-over-UDP Over IP-MPLS-IP Network

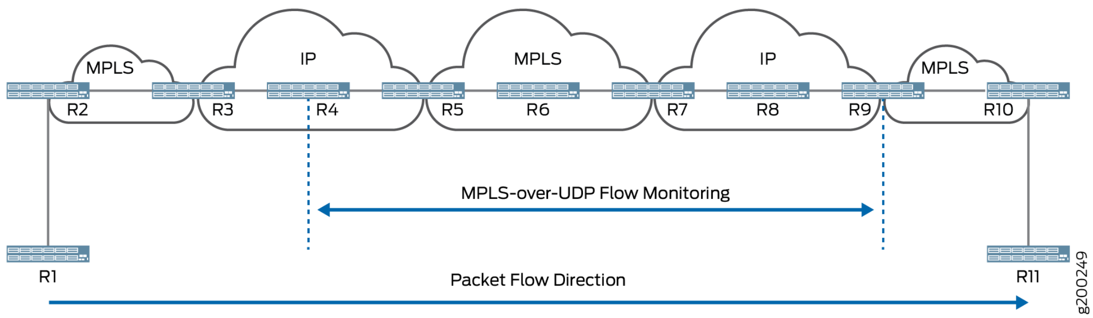


Figure 58: MPLS-over-UDP in RSVP-TE LSP Packet

Configuring Inline Active Flow Monitoring of MPLS-over-UDP Flows

IN THIS SECTION

(Junos OS only) Configuring inline active monitoring of MPLS-over-UDP flows includes the following tasks:

Configuring the Template to Specify Output Properties

Configure a template to specify the output properties for the flow records:

1. Configure the template name.

```
[edit services flow-monitoring]
user@host# set (version-ipfix | version9) template template-name
```

2. (Optional) Configure the interval after which an active flow is exported.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-active-timeout seconds
```

3. (Optional) Configure the interval of activity that marks a flow as inactive.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-inactive-timeout seconds
```

4. (Optional) Configure the frequency at which the flow generator sends updates about template definitions to the flow collector. Specify either number of packets or number of seconds.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set template-refresh-rate packets packets seconds seconds
```

5. (Optional) Configure the refresh rate in either number of packets or number of seconds.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set option-refresh-rate packets packets seconds seconds
```

6. Enable flow monitoring of MPLS-over-UDP flows.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set tunnel-observation mpls-over-udp
```

7. Specify the template type.

   - If you are monitoring an MPLS-over-UDP flow that is carried through a full IP network (see Figure 55 on page 570), use the `ipv4-template`:

   ```
   [edit services flow-monitoring (version-ipfix | version9) template template-name]
   user@host# set ipv4-template
   ```

- If you are monitoring an MPLS-over-UDP flow that is carried through an IP-MPLS-IP network (see Figure 57 on page 571):

  For the IP network transit and egress nodes (for example, R4, R5, R8, and R9 in Figure 57 on page 571), use the `ipv4-template` type.

  ```
  [edit services flow-monitoring (version-ipfix | version9) template template-name]
  user@host# set ipv4-template
  ```

  For the transit and egress nodes where the MPLS-over-UDP flow is encapsulated in an RSVP-TE LSP (for example R6 and R7 in Figure 57 on page 571), use one of the following templates:

  - Starting in Junos OS Release 18.2R1:

    ```
    [edit services flow-monitoring (version-ipfix | version9) template template-name]
    user@host# set mpls-template
    ```

  - In Junos OS Release 18.1:

    ```
    [edit services flow-monitoring (version-ipfix | version9) template template-name]
    user@host# set mpls-ipvx-template
    ```

8. Enable the learning of next-hop addresses so that the true outgoing interface (OIF) is reported.

   ```
   [edit services flow-monitoring (version-ipfix | version9) template template-name]
   user@host# set nexthop-learning
   ```

Configuring the Sampling Instance

Configure a sampling instance:

1. Configure the sampling instance name.

   ```
   [edit forwarding-options sampling]
   user@host# set instance instance-name
   ```

2. Configure the MPLS protocol family for the sampling instance.

```
[edit forwarding-options sampling instance instance-name]
user@host# set family mpls
```

3. Set the ratio of the number of packets to be sampled. For example, if you specify a rate of 10, every tenth packet (1 packet out of 10) is sampled.

```
[edit forwarding-options sampling instance instance-name input]
user@host# set rate number
```

4. Specify the source address for the traffic to be sampled.

```
[edit forwarding-options sampling instance instance-name family mpls output]
user@host# set inline-jflow source-address address
```

5. Specify the flow export rate of monitored packets in kpps.

```
[edit forwarding-options sampling instance instance-name family mpls output]
user@host# set inline-jflow flow-export-rate rate
```

6. Specify the output address and port for a flow server.

```
[edit forwarding-options sampling instance instance-name family mpls output]
user@host# set flow-server hostname port port-number
```

7. Specify the template to use with the sampling instance.

```
[edit forwarding-options sampling instance instance-name family mpls output flow-server
hostname]
user@host# set (version9 | version-ipfix) template template-name
```

Assigning the Sampling Instance to an FPC

- Assign the sampling instance to the FPC on which you want to implement flow monitoring.

```
[edit chassis]
user@host# set fpc slot-number sampling-instance instance-name
```

Configuring a Firewall Filter

Configure a firewall filter to accept and sample MPLS traffic.

1. Configure the MPLS firewall filter name.

```
[edit firewall]
user@host# edit family mpls filter filter-name
```

2. Configure a term to sample and accept MPLS packets.

```
[edit firewall family mpls filter filter-name]
user@host# set term term-name then accept
user@host# set term term-name then sample
```

Assigning the Firewall Filter to the Monitored Interface

- Assign the input firewall filter to the interface you want to monitor.

```
[edit interfaces]
user@host# set interface-name unit logical-unit-number family mpls filter input filter-name
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
21.2R1	Starting with Junos OS Release 21.2R1, the QFX10002-60C supports inline active flow monitoring for MPLS, MPLS-IPv4, MPLS-IPv6, and MPLS-over-UDP traffic. Both IPFIX and version 9 templates are supported.

19.4R1	Starting with Junos OS Release 19.4R1, on the PTX10002-60C, you can perform inline flow monitoring for MPLS-over-UDP flows to look past the tunnel header to sample and report on the inner payload at both the transit and egress nodes of the tunnel.
19.4R1	Starting with Junos OS Release 19.4R1, the PTX10002-60C supports inline active flow monitoring for MPLS, MPLS-IPv4, MPLS-IPv6, and MPLS-over-UDP traffic. Both IPFIX and version 9 templates are supported.
18.1R1	Starting with Junos OS Release 18.1R1 on PTX Series routers with an FPC3, PTX10K-LC1101, PTX10K-LC1102, or PTX1000 card, you can perform inline flow monitoring for MPLS-over-UDP flows to look past the tunnel header to sample and report on the inner payload at both the transit and egress nodes of the tunnel.

Inline Active Flow Monitoring on IRB Interfaces

SUMMARY

You can perform inline active flow monitoring for IPv4 and IPv6 traffic on the integrated routing and bridging (IRB) interfaces on PTX Series routers.

IN THIS SECTION

Overview

On PTX Series routers, you can perform inline active flow monitoring for IPv4 and IPv6 traffic on the integrated routing and bridging (IRB) interfaces. Both IPFIX and version 9 templates for the flow monitoring are supported. For a description of the fields included in the templates, see "Understand Inline Active Flow Monitoring" on page 439.

Understand Inline Active Flow Monitoring on IRB interfaces

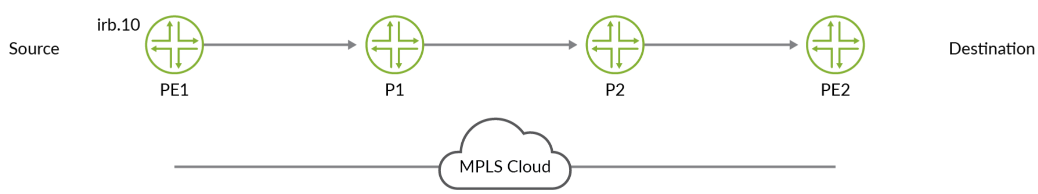
IN THIS SECTION

You can enable inline active flow monitoring by configuring the IPFIX or V9 templates on IRB interfaces.

Sampling on an IRB Interface with Traffic Routed to a Tunnelled Core

Figure 59 on page 577 illustrates sampling on an IRB interface where the traffic is routed to a tunnelled core, primarily an MPLS tunnel. The packets are entering irb.10 on which you can enable ingress sampling. The packets can be forwarded to a next hop which is not a part of any user-defined VLAN.

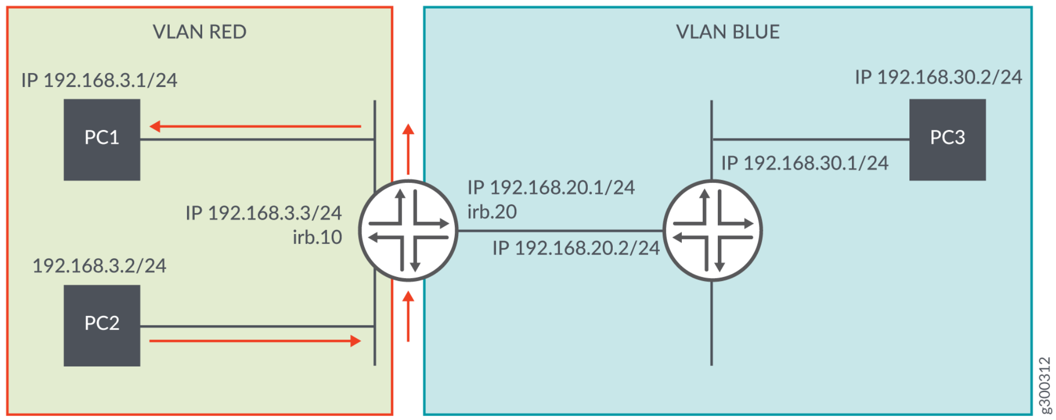
Figure 59: Sampling on an IRB Interface Routing Traffic to a Tunnelled Core



Layer 2 bridging and Layer 3 IP routing on an IRB interface

Figure 60 on page 578 illustrates the topology where Layer 2 bridging and Layer 3 IP routing are supported on the same interface.

Figure 60: Layer 2 Bridging and Layer 3 IP Routing on the Same IRB Interface

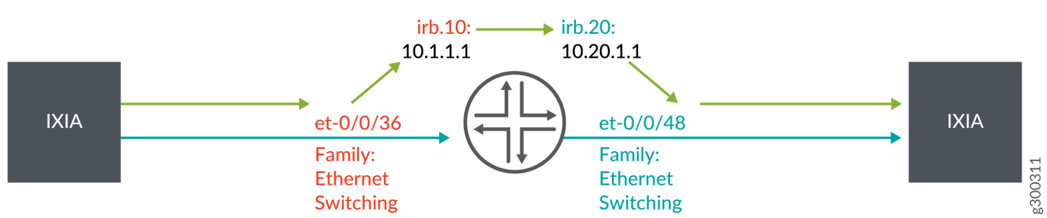


PC1 and PC2 are in VLAN RED (ID 10) and PC3 is in VLAN BLUE (ID 20).

For traffic moving from PC1 to PC3 or from PC2 to PC3, an IRB interface must be configured with a logical unit with an address in the subnet for VLAN RED and a logical unit with an address in the subnet for VLAN BLUE. The switch automatically directs routes to these subnets and uses these routes to forward traffic between VLANs. If traffic is flowing from VLAN RED to VLAN BLUE, you can configure ingress sampling on irb.10 and egress sampling on irb.20.

Figure 61 on page 578 illustrates sampling in a topology where Layer 2 bridging and Layer 3 IP routing are supported on the same interface. The interfaces, et-0/0/36.0 and irb.10 belong to VLAN ID 10. The interfaces, et-0/0/48 and irb.20 belong to VLAN ID 20. Packets are entering irb.10 and exiting on irb.20. Hence, you can configure ingress sampling on irb.10 and egress sampling on irb.20.

Figure 61: Sampling on an IRB Interface Supporting Bridging and Routing



Configure Inline Active Flow Monitoring on IRB Interfaces on PTX Series Routers

IN THIS SECTION

Configure the Template to Specify Output Properties

Configure a template to specify the output properties for the flow records:

1. Configure the template name.

```
[edit services flow-monitoring]
user@host# set (version-ipfix | version9) template template-name
```

For example:

```
[edit services flow-monitoring]
user@host# set version-ipfix template t1
```

2. (Optional) Configure the interval after which an active flow is exported.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-active-timeout seconds
```

For example:

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-active-timeout 10
```

3. (Optional) Configure the interval of activity that marks a flow as inactive.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-inactive-timeout seconds
```

For example:

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set flow-inactive-timeout 10
```

4. Specify the template type.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set template-name
```

For example:

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
user@host# set ipv4-template
```

Configure the Sampling Instance

Configure a sampling instance:

1. Configure the sampling instance name.

```
[edit forwarding-options sampling]
user@host# set instance instance-name
```

For example:

```
[edit forwarding-options sampling]
user@host# set instance s1
```

2. Configure the protocol family for the sampling instance.

```
[edit forwarding-options sampling instance instance-name]
user@host# set family (inet | inet6 | mpls)
```

For example:

```
[edit forwarding-options sampling instance instance-name]
user@host# set family inet
```

3. Set the ratio of the number of packets to be sampled. For example, if you specify a rate of 10, every tenth packet (1 packet out of 10) is sampled.

```
[edit forwarding-options sampling instance instance-name input]
user@host# set rate number
```

For example:

```
[edit forwarding-options sampling instance instance-name input]
user@host# set rate 10
```

4. Specify the source address for the traffic to be sampled.

```
[edit forwarding-options sampling instance instance-name family inet output]
user@host# set inline-jflow source-address address
```

For example:

```
[edit forwarding-options sampling instance instance-name family inet output]
user@host# set inline-jflow source-address 10.10.0.1
```

5. Specify the output address and port for a flow server.

```
[edit forwarding-options sampling instance instance-name family inet output]
user@host# set flow-server hostname port port-number
```

For example:

```
[edit forwarding-options sampling instance instance-name family inet output]
user@host# set flow-server 10.10.10.2 port 2055
```

6. Specify the template to use with the sampling instance.

```
[edit forwarding-options sampling instance instance-name family inet output flow-server
hostname]
user@host# set (version9 | version-ipfix) template template-name
```

For example:

```
[edit forwarding-options sampling instance instance-name family inet output]
user@host# set version-ipfix template t1
```

Assign the Sampling Instance to an FPC

Assign the sampling instance to the FPC on which you want to implement flow monitoring.

```
[edit chassis]
user@host# set fpc slot-number sampling-instance instance-name
```

For example:

```
[edit chassis]
user@host# set fpc 0 sampling-instance s1
```

Configure a Firewall Filter

Configure a firewall filter to specify the family of traffic to accept and sample.

1. Configure the firewall filter name and specify the family of traffic.

```
[edit firewall]
user@host# set family (inet | inet6 | mpls) filter filter-name
```

For example:

```
[edit firewall]
user@host# set family inet filter f2
```

2. Configure a term to sample and accept packets.

```
[edit firewall family mpls filter filter-name]
user@host# set term term-name then accept
user@host# set term term-name then sample
```

For example:

```
[edit firewall family mpls filter filter-name]
user@host# set term t1 then count c2
user@host# set term t1 then accept
user@host# set term t1 then sample
```

Associate a Layer 3 Interface with the VLAN to Route Traffic

Assign the IRB Interface to the VLAN.

```
[edit vlans vlan-name]
user@host# set vlan-name vlan-id vlan-id-number


user@host# set vlan-name l3-interface l3-interface-name

                     .logical-interface-number
```

For example:

```
[edit vlans vlan-name]
user@host# set vlan10 vlan-id 10
user@host# set vlan10 l3-interface irb.10
```

For example, if you are configuring inline flow monitoring using IRB while supporting layer 2 bridging and layer 3 IP routing on the same interface (See Figure 61 on page 578):

```
[edit vlans vlan-name]
user@host# set vlan-10 vlan-id 10
user@host# set vlan-10 l3-interface irb.10
user@host# set vlan-20 vlan-id 20
user@host# set vlan-20 l3-interface irb.20
```

Assign the Firewall Filter to the Monitored Interface

Assign the input firewall filter to the interface you want to monitor. Also, configure the VLANs for which the interface can carry traffic.

```
[edit interfaces]
user@host# set interface-name unit logical-unit-number family (inet | inet6 | mpls) filter input
filter-name
address
```

For example, if you are configuring inline flow monitoring using IRB while supporting layer 2 bridging and layer 3 IP routing on the same interface (See Figure 61 on page 578):

```
[edit interfaces]
user@host# set et-0/0/36 unit 0 family ethernet-switching vlan members vlan10
user@host# set et-0/0/48 unit 0 family ethernet-switching vlan members vlan20
user@host# set et-0/0/60 unit 0 family inet address 10.10.10.1
user@host# set irb unit 1 family inet filter input f2
user@host# set irb unit 1 family inet address 10.1.1.1
user@host# set irb unit 2 family inet address 10.20.1.1
user@host# set irb unit 1 family inet address 10.1.1.1
user@host# set irb unit 2 family inet filter output f2
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.4R1-EVO	Starting in Junos OS Evolved Release 24.4R1 on the PTX10002-36QDD router, you can perform inline active flow monitoring for IPv4 and IPv6 traffic on integrated routing and bridging (IRB) interfaces.
22.2R1-EVO	Starting in Junos OS Evolved Release 22.2R1 on the PTX10003 router, you can perform inline active flow monitoring for IPv4 and IPv6 traffic on integrated routing and bridging (IRB) interfaces.
21.3R1-EVO	Starting in Junos OS Evolved Release 21.3R1 on the PTX10001-36MR, PTX10004, and PTX10008 routers, you can perform inline active flow monitoring for IPv4 and IPv6 traffic on integrated routing and bridging (IRB) interfaces.
19.1R1	Starting in Junos OS Release 19.1R1, on PTX Series routers, you can perform inline active flow monitoring for IPv4 and IPv6 traffic on integrated routing and bridging (IRB) interfaces.

Example: Configuring Inline Active Flow Monitoring on MX Series and T4000 Routers

IN THIS SECTION

Configuration

CLI Quick Configuration

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

Configuring Template Properties

```
set services flow-monitoring version9 template template1 flow-active-timeout 120
set services flow-monitoring version9 template template1 flow-inactive-timeout 60
set services flow-monitoring version9 template template1 template-refresh-rate packets 100
set services flow-monitoring version9 template template1 template-refresh-rate seconds 600
set services flow-monitoring version9 template template1 option-refresh-rate packets 100
set services flow-monitoring version9 template template1 option-refresh-rate seconds 600
set services flow-monitoring version9 template template1 ipv4-template
set services flow-monitoring version-ipfix template template-v61 flow-active-timeout 150
set services flow-monitoring version-ipfix template template-v61 flow-inactive-timeout 100
set services flow-monitoring version-ipfix template template-v61 template-refresh-rate seconds 30
set services flow-monitoring version-ipfix template template-v61 ipv6-template
```

Configuring a Sampling Instance

```
set forwarding-options sampling instance instance-1 input rate 1
set forwarding-options sampling instance instance-1 family inet output flow-server 10.50.1.2
port 2055
set forwarding-options sampling instance instance-1 family inet output flow-server 10.50.1.2
version9 template template1
set forwarding-options sampling instance instance-1 family inet output inline-jflow source-
address 10.50.1.100
set forwarding-options sampling instance instance-1 family inet output inline-jflow flow-export-
rate 10
set forwarding-options sampling instance instance-1 family inet6 output flow-server 10.50.1.2
port 2055
set forwarding-options sampling instance instance-1 family inet6 output flow-server 10.50.1.2
version-ipfix template template-v61
set forwarding-options sampling instance instance-1 family inet6 output inline-jflow source-
address 10.50.1.110
set forwarding-options sampling instance instance-1 family inet6 output inline-jflow flow-export-
rate 6
```

Configuring FPC Parameters

```
set chassis fpc 0 sampling-instance instance-1
set chassis fpc 0 inline-services flow-table-size ipv4-flow-table-size 8
set chassis fpc 0 inline-services flow-table-size ipv6-flow-table-size 7
```

Configuring Firewall Filters

```
set firewall family inet filter inet-sample term t1 then sample
set firewall family inet filter inet-sample term t1 then accept
set firewall family inet6 filter inet6-sample term t1 then sample
set firewall family inet6 filter inet6-sample term t1 then accept
```

Configuring Interface Properties

```
set interfaces ge-0/0/4 unit 0 family inet filter input inet-sample
set interfaces ge-0/0/4 unit 0 family inet address 10.150.1.1/24
set interfaces ge-0/1/6 unit 0 family inet6 filter input inet6-sample
set interfaces ge-0/1/6 unit 0 family inet6 address 2001:db8:0:2::1/64
```

Step-by-Step Procedure

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

1. Configure the template properties for inline active flow monitoring.

```
[edit services flow-monitoring]
user@router1# set version9 template template1 ipv4-template
user@router1# set version9 template template1 flow-active-timeout 120
user@router1# set version9 template template1 flow-inactive-timeout 60
user@router1# set version9 template template1 template-refresh-rate packets 100
user@router1# set version9 template template1 option-refresh-rate packets 100
user@router1# set version-ipfix template template-v61 ipv6-template
user@router1# set version-ipfix template template-v61 flow-active-timeout 150
user@router1# set version-ipfix template template-v61 flow-inactive-timeout 100
user@router1# set version-ipfix template template-v61 template-refresh-rate seconds 30
user@router1# set version-ipfix template template-v61 option-refresh-rate seconds 30
```

2. Configure the sampling instance for inline active flow monitoring.

```
[edit forwarding-options sampling]
user@router1# set instance instance-1 input rate 1
user@router1# set instance instance-1 family inet output flow-server 10.50.1.2 port 2055
```

```
user@router1# set instance instance-1 family inet output flow-server 10.50.1.2 version9
template template1
user@router1# set instance instance-1 family inet output inline-jflow source-address
10.50.1.100
user@router1# set instance instance-1 family inet output inline-jflow flow-export-rate 10
user@router1# set instance instance-1 family inet6 output flow-server 10.50.1.2 port 2055
user@router1# set instance instance-1 family inet6 output flow-server 10.50.1.2 version-ipfix
template template-v61
user@router1# set instance instance-1 family inet6 output inline-jflow source-address
10.50.1.110
user@router1# set instance instance-1 family inet6 output inline-jflow flow-export-rate 6
```

NOTE: Until you complete the next step for associating the sampling instance with an FPC, the instance remains inactive and is marked `inactive` in the configuration.

3. Associate the sampling instance with the FPC on which you want to implement inline active flow monitoring, and also configure the hash table sizes.

NOTE: In Junos OS releases earlier than Release 12.1, the following conditions are applicable for supporting backward compatibility when you configure the IPv4 and IPv6 flow table sizes for inline active flow monitoring:

- If you do not configure the `flow-table-size` statement at the `[edit chassis fpc` *`slot-number`* `inline-services]` hierarchy level, fifteen 256K entries are allocated by default for the IPv4 flow table and one 1K entry is allocated by default for the IPv6 flow table on the Packet Forwarding Engine.

- If you configure the `ipv4-flow-table-size` *`size`* statement at the `[edit chassis fpc` *`slot-number`* `inline-services flow-table-size]` hierarchy level and do not configure the `ipv6-flow-table-size` *`size`* statement at the `[edit chassis fpc` *`slot-number`* `inline-services flow-table-size]` hierarchy level, the number of units of 256K entries that you configure for the IPv4 flow table is allocated. For the IPv6 flow table, a default size of one 1K entry is allocated on the Packet Forwarding Engine.

- If you do not configure the `ipv4-flow-table-size` *`size`* statement at the `[edit chassis fpc slot-number inline-services flow-table-size]` hierarchy level and if you configure the `ipv6-flow-table-size` *`size`* statement at the `[edit chassis fpc slot-number inline-services flow-table-size]` hierarchy level, the number of units of 256K entries that you configure for the IPv6 flow table is allocated. For the IPv4 flow table, a default size of one 1K entry is allocated on the Packet Forwarding Engine.

- If you configure the sizes of both the IPv4 and IPv6 flow tables, the flow tables are created on the Packet Forwarding Engine based on the size that you specified.

NOTE: When you configure inline active flow monitoring for VPLS flows, include the `vpls-flow-table-size` statement.

```
[edit chassis]
user@router1# set fpc 0 sampling-instance instance-1
user@router1# set fpc 0 inline-services flow-table-size ipv4-flow-table-size 8
user@router1# set fpc 0 inline-services flow-table-size ipv6-flow-table-size 7
```

4. Configure firewall filters.

```
[edit firewall]
user@router1# set family inet filter inet-sample term t1 then sample
user@router1# set family inet filter inet-sample term t1 then accept
user@router1# set family inet6 filter inet6-sample term t1 then sample
user@router1# set family inet6 filter inet6-sample term t1 then accept
```

5. Associate the firewall filters configured in the previous step with the interfaces on which you want to set up inline active flow monitoring.

```
[edit interfaces]
user@router1# set ge-0/0/4 unit 0 family inet filter input inet-sample
user@router1# set ge-0/0/4 unit 0 family inet address 10.150.1.1/24
user@router1# set ge-0/1/6 unit 0 family inet6 filter input inet6-sample
user@router1# set ge-0/1/6 unit 0 family inet6 address 2001:db8:0:2::1/64
```

6. Commit the configuration.

```
[edit]
user@router1# commit
```

Results

From the configuration mode, confirm your configuration by entering `show services flow-monitoring`, `show forwarding-options sampling`, `show chassis fpc 0`, `show firewall`, and `show interfaces` commands. If the output does not display the intended configuration, repeat the instructions in the example to correct the configuration.

- `show services flow-monitoring`

```
version9 {
    template template1 {
        flow-active-timeout 120;
        flow-inactive-timeout 60;
        template-refresh-rate {
            packets 100;
            seconds 600;
        }
        option-refresh-rate {
            packets 100;
            seconds 600;
        }
        ipv4-template;
    }
}
    version-ipfix {
        template template-v61 {
            flow-active-timeout 150;
            flow-inactive-timeout 100;
            template-refresh-rate {
                seconds 30;
            }
            ipv6-template;
        }
    }
```

- `show forwarding-options sampling`

```
instance {
    instance-1 {
        input {
            rate 1;
```

```
        }
        family inet {
            output {
                flow-server 10.50.1.2 {
                    port 2055;
                    version9 {
                        template {
                            template1;
                        }
                    }
                }
                inline-jflow {
                    source-address 10.50.1.100;
                    flow-export-rate 10;
                }
            }
        }
        family inet6 {
            output {
                flow-server 10.50.1.2 {
                    port 2055;
                    version-ipfix {
                        template {
                            template-v61;
                        }
                    }
                }
                inline-jflow {
                    source-address 10.50.1.110;
                    flow-export-rate 6;
                }
            }
        }
    }
}
```

- show chassis fpc 0

```
sampling-instance instance-1;
    inline-services {
        flow-table-size {
            ipv4-flow-table-size 8;
```

```
            ipv6-flow-table-size 7;
        }
    }
```

- show firewall

```
family inet {
    filter inet-sample {
        term t1 {
            then {
                sample;
                accept;
            }
        }
    }
}
    family inet6 {
        filter inet6-sample {
            term t1 {
                then {
                    sample;
                    accept;
                }
            }
        }
    }
```

- show interfaces

```
...
ge-0/1/6 {
    vlan-tagging;
    unit 0 {
        family inet6 {
            filter {
                input inet6-sample;
            }
            address 2001:db8:0:2::1/64;
        }
    }
    }
```

```
    ge-0/0/4 {
        vlan-tagging;
        unit 0 {
            family inet {
                filter {
                    input inet-sample;
                }
                address 10.150.1.1/24;
            }
        }
    }
...
```

Software and Hardware Requirements

- An MX Series router other than MX80
- Junos OS Release 13.2 or later.

NOTE:

- Junos OS Releases earlier than 13.2 also support inline active flow monitoring. However, some of the features discussed in this example are not supported on previous releases.
- You need Junos OS Release 14.2 or later for configuring inline active flow monitoring on T4000 routers with Type 5 FPC.

Overview

Inline active flow monitoring enables you to configure active sampling without making use of a services DPC. This topic explains the basic configuration for enabling inline active flow monitoring for IPv4 and IPv6 flows. You can also configure inline active flow monitoring for VPLS flows. To configure inline active flow monitoring for VPLS flows, you must specify the `family` as `vpls` and include `vpls-template` at the `[edit services flow-monitoring version-ipfix template` *`template-name`*`]` hierarchy level.

RELATED DOCUMENTATION

Understand Inline Active Flow Monitoring | 439

Configuring Inline Active Flow Monitoring on MX80 and MX104 Routers | 555

CHAPTER 14

Sampling Data Using Flow Aggregation

IN THIS CHAPTER

Understanding Flow Aggregation

You can collect an aggregate of sampled flows and send the aggregate to a specified host that runs either the cflowd application available from CAIDA (http://www.caida.org) or the newer version 9 format defined in RFC 3954, *Cisco Systems NetFlow Services Export Version 9*. Before you can perform flow aggregation, the routing protocol process must export the autonomous system (AS) path and routing information to the sampling process.

By using flow aggregation, you can obtain various types of byte and packet counts of flows through a router. The application collects the sampled flows over a period of 1 minute. At the end of the minute, the number of samples to be exported are divided over the period of another minute and are exported over the course of the same minute.

You configure flow aggregation in different ways, depending on whether you want to export flow records in cflowd version 5 or 8 format, or the separate version 9 format. The latter allows you to sample MPLS, IPv4, IPv6, and peer AS billing traffic. You can also combine configuration statements between the MPLS and IPv4 formats.

> **NOTE**: When PIC-based sampling is enabled, collection of flow statistics for sampled packets on flows in virtual private networks (VPNs) is also supported. No additional CLI configuration is required.

RELATED DOCUMENTATION

Enabling Flow Aggregation | 595

Configuring Flow Aggregation on MX, M and T Series Routers and NFX250 to Use Version 5 or Version 8 cflowd | 596

Configuring Flow Aggregation on MX, M, vMX and T Series Routers and NFX250 to Use Version 9 Flow Templates | 601

Directing Replicated Flows from M and T Series Routers to Multiple Flow Servers | 658

Logging cflowd Flows on M and T Series Routers Before Export | 661

Enabling Flow Aggregation

Before you can perform flow aggregation, the routing protocol process must export the autonomous system (AS) path and routing information to the sampling process. To enable the export of AS path and the routing information to the sampling process, one or more of the following needs to be configured:

- At the `[edit forwarding-options]` hierarchy level (for routing instances, at the `[edit routing-instance` *`routing-instance-name`* `forwarding-options]` hierarchy level), configure `sampling family` or `sampling output` or `sampling instance` or `monitoring` or `accounting`.
- At the `[edit routing-options]` hierarchy level (for routing instances, at the `[edit routing-instance` *`routing-instance-name`* `routing-options]` hierarchy level), configure `route record`.
- At the `[edit chassis fpc` *`slot-number`* `pic` *`pic-number`* `adaptive-services service-package extension-provider]` hierarchy level, configure `forwarding-db-size`.

RELATED DOCUMENTATION

Understanding Flow Aggregation | 594

Configuring Flow Aggregation on MX, M and T Series Routers and NFX250 to Use Version 5 or Version 8 cflowd | 596

Configuring Flow Aggregation on MX, M, vMX and T Series Routers and NFX250 to Use Version 9 Flow Templates | 601

Directing Replicated Flows from M and T Series Routers to Multiple Flow Servers | 658

Configuring Traffic Sampling on MX, M and T Series Routers | 422

Logging cflowd Flows on M and T Series Routers Before Export | 661

Configuring Flow Aggregation on MX, M and T Series Routers and NFX250 to Use Version 5 or Version 8 cflowd

To enable the collection of cflowd version 5 or version 8 flow formats, include the `flow-server` statement:

```
flow-server hostname {
    aggregation {
        autonomous-system;
        destination-prefix;
        protocol-port;
        source-destination-prefix {
            caida-compliant;
        }
        source-prefix;
    }
    autonomous-system-type (origin | peer);
    (local-dump | no-local-dump);
    port port-number;
    version format;
}
```

You can include this statement at the following hierarchy levels:

- `[edit forwarding-options sampling family (inet | inet6 | mpls) output]`
- `[edit forwarding-options sampling instance instance-name output]`
- `[edit forwarding-options accounting name output cflowd hostname]`

You must configure the `family inet` statement on logical interface `unit 0` on the monitoring interface, as in the following example:

```
[edit interfaces]
sp-3/0/0 {
    unit 0 {
        family inet {
            ...
        }
    }
}
```

NOTE: Boot images for monitoring services interfaces are specified at the `[edit chassis images pic]` hierarchy level. You must enable the NTP client to make the cflowd feature operable, by including the following configuration:

```
[edit system]
ntp {
    boot-server ntp.example.com;
    server 172.17.28.5;
}
processes {
    ntp enable;
}
```

You can also configure cflowd version 5 for flow-monitoring applications by including the `cflowd` statement at the `[edit forwarding-options monitoring` *`name`* `family inet output]` hierarchy level:

```
cflowd hostname {
    port port-number;
}
```

The following restrictions apply to cflowd flow formats:

- You can configure up to one version 5 and one version 8 flow format at the `[edit forwarding-options accounting` *`name`* `output]` hierarchy level.
- You can configure up to eight version 5 or one version 8 flow format at the `[edit forwarding-options sampling family (inet | inet6 | mpls) output]` hierarchy level for Routing Engine-based sampling by including the `flow-server` statement. In contrast, PIC-based sampling allows you to specify one cflowd

version 5 server and one version 8 server simultaneously. However, the two cflowd servers must have different IP addresses.

- You can configure up to eight version 5 flow formats at the `[edit forwarding-options monitoring name output]` hierarchy level. Version 8 flow formats and aggregation are not supported for flow-monitoring applications.

- Outbound Routing Engine traffic is not sampled. A firewall filter is applied as output on the egress interface, which samples packets and exports the data. For transit traffic, egress sampling works correctly. For internal traffic, the next hop is installed in the Packet Forwarding Engine but sampled packets are not exported.

- Flows are created on the monitoring PIC only after the route record resynchronization operation is complete, which is 60 seconds after the PIC comes up. Any packets sent to the PIC are dropped until the synchronization process is complete.

- The configuration includes a proprietary v5 extension template for supporting 4-byte AS information in flow records. Its template version is set to 500, indicating it to be proprietary. All other fields remain the same; the source AS and destination AS are each 4 bytes long, rather than 2 bytes as in the traditional v5 template. This option is available at the `[edit forwarding-options sampling family inet output flow-server server-name version]` hierarchy level.

In the `cflowd` statement, specify the name or identifier of the host that collects the flow aggregates. You must also include the User Datagram Protocol (UDP) port number on the host and the version, which gives the format of the exported cflowd aggregates. To collect cflowd records in a log file before exporting, include the `local-dump` statement.

NOTE: You can specify both host (cflowd) sampling and port mirroring in the same configuration; however, only one action takes effect at any one time. Port mirroring takes precedence. For more information, see Configuring Port Mirroring on M, T MX, ACX, and PTX Series Routers.

For cflowd version 8 only, you can specify aggregation of specific types of traffic by including the `aggregation` statement. This conserves memory and bandwidth by enabling cflowd to export targeted flows rather than all aggregated traffic. To specify a flow type, include the `aggregation` statement:

```
aggregation {
    autonomous-system;
    destination-prefix;
    protocol-port;
    source-destination-prefix {
        caida-compliant;
    }
```

```
    source-prefix;
}
```

You can include this statement at the following hierarchy levels:

- `[edit forwarding-options sampling family (inet | inet6 | mpls) output flow-server` *`hostname`*`]`
- `[edit forwarding-options accounting` *`name`* `output cflowd` *`hostname`*`]`

The `autonomous-system` statement configures aggregation by the AS number; this statement might require setting the separate cflowd `autonomous-system-type` statement to include either `origin` or `peer` AS numbers. The `origin` option specifies to use the origin AS of the packet source address in the Source Autonomous System cflowd field. The `peer` option specifies to use the peer AS through which the packet passed in the Source Autonomous System cflowd field. By default, cflowd exports the origin AS number.

The `destination-prefix` statement configures aggregation by the destination prefix only.

The `protocol-port` statement configures aggregation by the protocol and port number; requires setting the separate `cflowd port` statement.

The `source-destination-prefix` statement configures aggregation by the source and destination prefix. Version 2.1b1 of CAIDA's cflowd application does not record source and destination mask length values in compliance with CAIDA's *cflowd Configuration Guide*, dated August 30, 1999. If you configure the `caida-compliant` statement, the Junos OS complies with Version 2.1b1 of cflowd. If you do not include the `caida-compliant` statement in the configuration, the Junos OS records source and destination mask length values in compliance with the *cflowd Configuration Guide*.

The `source-prefix` statement configures aggregation by the source prefix only.

Collection of sampled packets in a local ASCII file is not affected by the `cflowd` statement.

The following commands enable Routing Engine- and PIC-based sampling at the `set forwarding options sampling` hierarchy level:

- `set input rate` *`rate`*
- `set input run-length` *`length`*
- `set family inet output flow-server` *`flowcollector`* `port` *`udp port`*
- `set family inet output flow-server` *`flowcollector`* `no-local-dump`
- `set family inet output flow-server` *`flowcollector`* `version <5/8>`

The following commands enable Routing Engine- and PIC-based sampling at the `set interfaces` hierarchy level:

- *interface to be sampled* unit *unit* family inet filter *input/output filtername*

The following commands enable Routing Engine- and PIC-based sampling at the `set firewall family` hierarchy level:

- `set inet filter` *`filtername`* `term 1 then count` *`filtername`*`ing`
- `set inet filter` *`filtername`* `term 1 then sample`
- `set inet filter` *`filtername`* `term 1 then accept`

The following command enables PIC-based sampling at the `set forwarding options sampling` hierarchy level:

- `set family inet output interface` *`sp-*/*/*`* `source address` *`source address`*

The following example shows a PIC-based flow aggregation configuration using version 5:

```
family inet {
    output {
        flow-inactive-timeout 15;
        flow-active-timeout 60;
        flow-server 203.0.113.165 {
            port 9996;
            version 5;
        }
        interface sp-2/2/0 {
        engine-id 4;
        source-address 203.0.113.126;
    }

}
```

The following example shows an Routing Engine-based flow aggregation configuration using version 5:

```
family inet {
    output {
        flow-inactive-timeout 15;
        flow-active-timeout 60;
        flow-server 203.0.113.165 {
            port 9996;
            source-address 203.0.113.126;
            version 5;
        }
    }
```

RELATED DOCUMENTATION

Configuring Flow Aggregation on MX, M, vMX and T Series Routers and NFX250 to Use Version 9 Flow Templates

IN THIS SECTION

Use of version 9 flow template enables you to define a flow record template suitable for IPv4 traffic, IPv6 traffic, MPLS traffic, a combination of IPv4 and MPLS traffic, or peer AS billing traffic. Templates and the fields included in the template are transmitted to the collector periodically, and the collector does not affect the router configuration.

NOTE: Version 9 requires that you install a services PIC, such as the Adaptive Services PIC or MS-PIC in the router. On MX Series routers, the MS-DPC fulfills this requirement. For more information on determining which services PIC is suitable for your router, see *Enabling Service Packages* or the appropriate hardware documentation.

NOTE: If multiple protocol families are configured for a particular flow collector, the export packets originates from multiple Source IDs, with each Source ID corresponding to a particular protocol. The multiple Source IDs do not indicate that the export packets are originating from multiple Service PICs.

Configuring the Traffic to Be Sampled

To specify sampling of IPv4, IPv6, MPLS, or peer AS billing traffic, include the appropriate configuration of the `family` statement at the `[edit forwarding-options sampling]` hierarchy level:

```
[edit forwarding-options]
sampling {
family (inet | inet6 | mpls);
}
```

You can include `family inet` ,`family inet6`, or `family mpls`.

NOTE: If you specify sampling for peer AS billing traffic, the `family` statement supports only IPv4 and IPv6 traffic (inet or inet6). Peer AS billing traffic is enabled only at the global instance hierarchy level and is not available for per Packet Forwarding Engine instances.

After you specify the family of traffic to be sampled, configure the sampling parameters such as:

- Maximum packet length (beyond which the packets are truncated).
- Maximum packets to be sampled per second (beyond which the packets are dropped).
- Rate (for example, if you specify 10, every 10th packet is sampled).
- Run length (which specifies the number of packets to be sampled after the trigger; that is, if the `rate` is set to 10 and `run-length` to 5, five packets starting at the 10th packet are sampled).

```
[edit forwarding-options sampling]
input {
maximum-packet-length bytes
max-packets-per-second number;
rate number;
```

```
run-length number;
}
```

Configuring the Version 9 Template Properties

To define the Version 9 templates, include the following statements at the `[edit services flow-monitoring version9]` hierarchy level:

```
[edit services flow-monitoring version9]
template template-name {
    options-template-id
    template-id
    source-id
    flow-active-timeout seconds;
    flow-inactive-timeout seconds;
    option-refresh-rate packets packets seconds seconds;
    template-refresh-rate packets packets seconds seconds;
    (ipv4-template | ipv6-template | mpls-ipv4-template | mpls-template | peer-as-billing-template) {
        label-position [ positions ];
    }
}
```

The following details apply to the configuration statements:

- You assign each template a unique name by including the `template` *`template-name`* statement.
- You then specify each template for the appropriate type of traffic by including the `ipv4-template`, `ipv6-template`, `mpls-ipv4-template`, or `mpls-template`.
- If the template is used for MPLS traffic, you can also specify up to three label positions for the MPLS header label data by including the `label-position` statement; the default values are `[1 2 3]`.
- Within the template definition, you can optionally include values for the `flow-active-timeout` and `flow-inactive-timeout` statements. These statements have specific default and range values when they are used in template definitions; the default is 60 seconds and the range is from 10 through 600 seconds. Values you specify in template definitions override the global timeout values configured at the `[edit forwarding-options sampling family (inet | inet6 | mpls) output flow-server]` hierarchy level.
- You can also include settings for the `option-refresh-rate` and `template-refresh-rate` statements within a template definition. For both of these properties, you can include a timer value (in seconds) or a packet count (in number of packets). For the `seconds` option, the default value is 60 and the range is

from 10 through 600. For the `packets` option, the default value is 4800 and the range is from 1 through 480,000.

- To filter IPv6 traffic on a media interface, the following configuration is supported:

```
interfaces interface-name {
    unit 0 {
        family inet6 {
            sampling {
                input;
                output;
            }
        }
    }
}
```

Customizing Template ID, Observation Domain ID, and Source ID for Version 9 Flow Templates

Starting in Junos OS Release 14.1, you can define a Version 9 flow record template suitable for IPv4 traffic, IPv6 traffic, MPLS traffic, a combination of IPv4 and MPLS traffic, or peer AS billing traffic. Templates and the fields included in the template are transmitted to the collector periodically,and the collector does not affect the router configuration. You can specify the unique identifier for the version 9 and IPFIX templates. The identifier of a template is locally unique within a combination of a transport session and an observation domain. Template IDs 0 through 255 are reserved for template sets, options template sets, and other sets for future use. Template IDs of data sets are numbered from 256 through 65535. Typically, this information element or field in the template is used to define the characteristics or properties of other information elements in a template. After a restart of the export process of templates is performed, you can reassign template IDs.

This functionality to configure template ID, options template ID, observation domain ID, and source ID is supported on all routers with MPCs.

NOTE: The template IDs that include MPLS and MPLS-IPv4 template ID are applicable for IPFIX only. The V9 format carries a different template ID.

The corresponding data sets and option data sets contain the value of the template IDs and options template IDs respectively in the set ID field. This method enables the collector to match a data record with a template record.

For more information about specifying the source ID, observation domain ID, template ID, and options template ID for version 9 and IPFIX flows, see "Configuring Observation Domain ID and Source ID for Version 9 and IPFIX Flows" on page 638 and "Configuring Template ID and Options Template ID for Version 9 and IPFIX Flows" on page 643.

Restrictions

The following restrictions apply to version 9 templates:

- You cannot apply the two different types of flow aggregation configuration at the same time.
- Flow export based on an `mpls-ipv4` template assumes that the IPv4 header follows the MPLS header. In the case of Layer 2 VPNs, the packet on the provider router (P router) looks like this:

  ```
  MPLS | Layer 2 Header | IPv4
  ```

  In this case, `mpls-ipv4` flows are not created on the PIC, because the IPv4 header does not directly follow the MPLS header. Packets are dropped on the PIC and are accounted as parser errors.
- Outbound Routing Engine traffic is not sampled. A firewall filter is applied as output on the egress interface, which samples packets and exports the data. For transit traffic, egress sampling works correctly. For internal traffic, the next hop is installed in the Packet Forwarding Engine but sampled packets are not exported.
- Flows are created on the monitoring PIC only after the route record resynchronization operation is complete, which is 60 seconds after the PIC comes up. Any packets sent to the PIC are dropped until the synchronization process is complete.

> **NOTE**: Because the forwarding of a packet that arrives with MPLS labels is performed based on the MPLS label and not based on the IP address contained in the packet, the packet is sampled at the output interface with the MPLS label that was popped not being available at the time of sampling. In such a case, depending on the incoming interface (IIF), the VRF index is identified and the route for the sampled packet is determined in the VRF table. Because a specific route is not available in the VRF that is different from the VRF on which the packet is received, the Output Interface Index, Source Mask, and Destination Mask fields are incorrectly populated. This behavior occurs when an IPv4 template is applied as a firewall filter on an egress interface with sample as the action.

Fields Included in Each Template Type

The following fields are common to all template types:

- Input interface
- Output interface
- Number of bytes
- Number of packets
- Flow start time
- Flow end time

The IPv4 template includes the following specific fields:

- IPv4 Source Address
- IPv4 Destination Address
- L4 Source Port
- L4 Destination Port
- IPv4 ToS
- IPv4 Protocol
- ICMP type and code
- TCP Flags
- IPv4 Next Hop Address
- Source autonomous system (AS) number
- Destination AS number

The IPv6 template includes the following specific fields:

- IPv6 Source Address and Mask
- IPv6 Destination Address and Mask
- L4 Source Port
- L4 Destination Port
- IPv6 ToS
- IPv6 Protocol
- TCP Flags

- IP Protocol Version
- IPv6 Next Hop Address
- Egress Interface Information
- Source Autonomous System (AS) number
- Destination AS number

The MPLS template includes the following specific fields:

- MPLS Label #1
- MPLS Label #2
- MPLS Label #3
- MPLS EXP Information
- FEC IP Address

The MPLS-IPv4 template includes all the fields found in the IPv4 and MPLS templates.

The peer AS billing template includes the following specific fields:

- IPv4 Class of Service (ToS)
- Ingress Interface
- BGP IPv4 Next Hop Address
- BGP Peer Destination AS Number

MPLS Sampling Behavior

This section describes the behavior when MPLS sampling is used on egress interfaces in various scenarios (label pop or swap) on provider routers (P routers). For more information on configuration and background specific to MPLS applications, see the MPLS Applications User Guide.

- You configure MPLS sampling on an egress interface on the P router and configure an MPLS flow aggregation template. The route action is label *pop* because penultimate hop popping (PHP) is enabled.

  With the current capability of applying MPLS templates, MPLS flows are created.

- As in the first case, you configure MPLS sampling on an egress interface on the P router and configure an MPLS flow aggregation template. The route action is label swap and the swapped label is 0 (explicit null).

The resulting behavior is that MPLS packets are sent to the PIC. The flow being sampled corresponds to the label before the swap.

- You configure a Layer 3 VPN network, in which a customer edge router (CE-1) sends traffic to a provider edge router (PE-A), through the P router, to a similar provider edge router (PE-B) and customer edge router (CE-2) on the remote end.

  The resulting behavior is that you cannot sample MPLS packets on the PE-A to P router link.

Verification

To verify the configuration properties, you can use the `show services accounting aggregation template template-name` *`name`* operational mode command.

All other `show services accounting` commands also support version 9 templates, except for `show services accounting flow-detail` and `show services accounting aggregation aggregation-type`. For more information about operational mode commands, see the CLI Explorer.

Examples: Configuring Version 9 Flow Templates

The following example shows a version 9 template configuration:

```
services {
    flow-monitoring {
        version9 {
            template ip-template {
                flow-active-timeout 20;
                flow-inactive-timeout 120;
                ipv4-template;
            }
            template mpls-template-1 {
                mpls-template {
                    label-position [1 3 4];
                }
            }
            template mpls-ipv4-template-1 {
                mpls-ipv4-template {
                    label-position [1 5 7];
                }
            }
            template vpls-template-1 {
                vpls-template;
                }
```

```
            }
        }
    }
}
```

The following example shows a firewall filter configuration for MPLS traffic:

```
firewall {
    family mpls {
        filter mpls_sample {
            term default {
                then {
                    accept;
                    sample;
                }
            }
        }
    }
}
```

The following example applies the MPLS sampling filter on a networking interface and configures the AS PIC to accept both IPv4 and MPLS traffic:

```
interfaces {
    at-0/1/1 {
        unit 0 {
            family mpls {
                filter {
                    input mpls_sample;
                }
            }
        }
    }
    sp-7/0/0 {
        unit 0 {
            family inet;
            family mpls;
        }
    }
}
```

The following example applies the MPLS version 9 template to the sampling output and sends it to the AS PIC:

```
forwarding-options {
    sampling {
        input {
            family mpls {
                rate 1;
            }
        }
        family mpls {
            output {
            flow-active-timeout 60;
            flow-inactive-timeout 30;
            flow-server 192.0.2.4 {
                port 2055;
                version9 {
                    template mpls-ipv4-template-1;
                }
            }
            interface sp-7/0/0 {
                source-address 198.51.100.1;
            }
            }
        }
    }
}
```

The following example shows a firewall filter configuration for the peer AS billing traffic:

```
firewall {
    family inet {
        filter peer-as-filter {
            term 0 {
                    from {
                        destination-class dcu-1;
                        interface ge-2/1/0;
                        forwarding-class class-1;
                    }
                    then count count_team_0;
            }
```

```
            }
            term 1 {
                    from {
                        destination-class dcu-2;
                        interface ge-2/1/0;
                        forwarding-class class-1;
                    }
                    then count count_team_1;
            }
            term 2 {
                    from {
                        destination-class dcu-3;
                        interface ge-2/1/0;
                        forwarding-class class-1;
                    }
                    then count count_team_2;
            }
        }
    }
}
```

The following example applies the peer AS firewall filter as a filter attribute under the forwarding-options hierarchy for CoS-level data traffic usage information collection:

```
forwarding-options {
    family inet {
        filter output peer-as-filter;
    }
}
```

The following example applies the peer AS DCU policy options to collect usage statistics for the traffic stream for as-path ingressing at a specific input interface with the firewall configuration hierarchy applied as Forwarding Table Filters (FTFs). The configuration functionality with CoS capability can be achieved through FTFs for destination-class usage with forwarding-class for specific input interfaces:

```
policy-options {
    policy-statement P1 {
            from {
                protocol bgp;
                neighbor 10.2.25.5; #BGP router configuration;
                as-path AS-1; #AS path configuration;
```

```
            }
            then destination-class dcu-1; #Destination class configuration;
    }
    policy-statement P2 {
            from {
                neighbor 203.0.113.5;
                as-path AS-2;
            }
            then destination-class dcu2;
    }
    policy-statement P3 {
            from {
                protocol bgp;
                neighbor 192.0.2.129;
                as-path AS-3;
            }
            then destination-class dcu3;
    }
    as-path AS-1 3131:1111:1123;
    as-path AS-2 100000;
    as-path AS-3 192:29283:2;
}
```

The following example applies the vpls version 9 template to enable sampling of traffic for billing purposes:

```
forwarding-options {
    sampling {
        }
        input {
            rate 1;
        }
        family inet {
            output {
                flow-server 10.209.15.58 {
                    port 300;
                    version9 {
                        template {
                            peer-as;
                        }
                    }
                }
```

```
                interface sp-5/2/0 {
                    source-address 203.0.113.133;
                }
            }
        }
    }
}
family inet {
    filter {
        output peer-as-filter;
    }
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can define a Version 9 flow record template suitable for IPv4 traffic, IPv6 traffic, MPLS traffic, a combination of IPv4 and MPLS traffic, or peer AS billing traffic.

RELATED DOCUMENTATION

Understanding Flow Aggregation | 594

Enabling Flow Aggregation | 595

Configuring Flow Aggregation on MX, M and T Series Routers and NFX250 to Use Version 5 or Version 8 cflowd | 596

Configuring Inline Active Flow Monitoring to Use IPFIX Flow Templates on MX, vMX and T Series Routers, EX Series Switches, NFX Series Devices, and SRX Series Firewalls | 621

Configuring Traffic Sampling on MX, M and T Series Routers | 422

Configuring Flow Aggregation on PTX Series Routers to Use Version 9 Flow Templates

IN THIS SECTION

You can define a flow record template suitable for IPv4 traffic or IPv6 traffic using a version 9 flow template. Templates and the fields included in the template are transmitted to the collector periodically. The collector does not affect the router configuration. You can define template refresh rate, flow active timeout and inactive timeout.

If flow records are being sent for multiple protocol families (for example, for IPv4 and IPv6), each protocol family flow will have a unique Observation Domain ID.

Configuring the Version 9 Template Properties

To define the version 9 templates, include the following statements at the `[edit services flow-monitoring version9]` hierarchy level:

```
[edit services flow-monitoring version9]
template name {
    options-template-id
    template-id
    observation-domain-id
    flow-active-timeout seconds;
    flow-inactive-timeout seconds;
    option-refresh-rate packets packets seconds seconds;
    template-refresh-rate packets packets seconds seconds;
```

```
    (ipv4-template | ipv6-template);
}
```

The following details apply to the configuration statements:

- You assign each template a unique name by including the `template` *`name`* statement.
- You specify each template for the appropriate type of traffic by including the `ipv4-template` or `ipv6-template`.
- Within the template definition, you can optionally include values for the `flow-active-timeout` and `flow-inactive-timeout` statements. These statements have specific default and range values when they are used in template definitions; the default is 60 seconds and the range is from 10 through 600 seconds.
- You can also include settings for the `option-refresh-rate` and `template-refresh-rate` statements within a template definition. For both of these properties, you can include a timer value (in seconds) or a packet count (in number of packets). For the `seconds` option, the default value is 600 and the range is from 10 through 600. For the `packets` option, the default value is 4800 and the range is from 1 through 480,000.
- To filter IPv6 traffic on a media interface, the following configuration is supported:

```
interfaces interface-name {
    unit 0 {
        family inet6 {
            sampling {
                input;
                output;
            }
        }
    }
}
```

Restrictions

The following restrictions apply to version 9 templates:

- Outbound Routing Engine traffic is not sampled. A firewall filter is applied as output on the egress interface, which samples packets and exports the data. For transit traffic, egress sampling works correctly. For internal traffic, the next hop is installed in the Packet Forwarding Engine but sampled packets are not exported.

- Flows are created only after the route record resynchronization operation is complete, which takes 120 seconds.

Customizing Template ID, Observation Domain ID, and Source ID for Version 9 flow Templates

> **NOTE**: For PTX Series routers with third generation FPCs installed, the FPC's slot number is used for the observation domain ID.

Use of version 9 flow templates allow you to define a flow record template suitable for IPv4 traffic or IPv6 traffic. Templates and the fields included in the template are transmitted to the collector periodically, and the collector does not need to be aware of the router configuration. Template IDs 0 through 255 are reserved for template sets, options template sets, and other sets for future use. Template IDs of data sets are numbered from 256 through 65535. Typically, this information element or field in the template is used to define the characteristics or properties of other information elements in a template. After a restart of the export process of templates is performed, template IDs can be reassigned.

The corresponding data sets and option data sets contain the value of the template IDs and options template IDs respectively in the set ID field. This method enables the collector to match a data record with a template record.

Fields Included in the IPv4 Templates for PTX Series Routers

Table 115 on page 616 shows the fields that are available in the IPv4 templates.

Table 115: IPv4 Template Fields

Field	Element ID
IPv4 Source Address	8
IPv4 Destination Address	12
IPv4 TOS	5
IPv4 Protocol	4

Table 115: IPv4 Template Fields *(Continued)*

Field	Element ID
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10
Source AS	16
Destination AS	17
BGP Next Hop Address	18
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv4 Next Hop Address	15
IPv4 Source Mask	9
IPv4 Destination Mask	13

Table 115: IPv4 Template Fields *(Continued)*

Field	Element ID
TCP Flags	6
IP Protocol Version	60

Fields Included in the IPv6 Templates for PTX Series Routers

Table 116 on page 618 shows the fields that are available in the IPv6 templates.

Table 116: IPv6 Template Fields

Field	Element ID
IPv6 Source Address	27
IPv6 Destination Address	28
IPv6 TOS	5
IPv6 Protocol	4
L4 Source Port	7
L4 Destination Port	11
ICMP Type and Code	32
Input Interface	10
Source AS	16
Destination AS	17

Table 116: IPv6 Template Fields *(Continued)*

Field	Element ID
Output Interface	14
Number of Flow Bytes	1
Number of Flow Packets	2
Time the flow started with respect to system up time (FPC up time)	22
Time the flow ended with respect to system up time (FPC up time)	21
IPv6 Next Hop Address	62
IPv6 Source Mask	29
IPv6 Destination Mask	30
TCP Flags	6
IP Protocol Version	60

Verification

The following show commands are supported for version 9:

- `show services accounting flow inline-jflow fpc-slot` *`fpc-slot`*
- `show services accounting errors inline-jflow fpc-slot` *`fpc-slot`*
- `show services accounting status inline-jflow fpc-slot` *`fpc-slot`*

Example: Configuring an version 9 Flow Templates and Flow Sampling

The following is a sample version 9 template configuration:

```
services {
    flow-monitoring {
        version9 {
            template ipv4 {
                flow-active-timeout 60;
                flow-inactive-timeout 70;
                template-refresh-rate seconds 30;
                option-refresh-rate seconds 30;
                ipv4-template;
            }
        }
    }
}
```

```
chassis;
    fpc 0 {
        sampling-instance s1;
    }
```

The following example applies the version 9 template to enable sampling of traffic for billing:

```
forwarding-options {
    sampling {
        instance {
            s1 {
                input {
                    rate 10;
                }
                family inet {
                    output {
                        flow-server 11.11.4.2 {
                            port 2055;
                            version9 {
                                template {
                                    ipv4;
                                }
```

```
                        }
                    }
                    inline-jflow {
                        source-address 11.11.2.1;
                    }
                }
            }
        }
    }
}
}
```

RELATED DOCUMENTATION

Configuring Inline Active Flow Monitoring on ACX and PTX Series Routers | 559

version9 (Flow Monitoring)

ipv4-template

ipv6-template

Configuring Inline Active Flow Monitoring to Use IPFIX Flow Templates on MX, vMX and T Series Routers, EX Series Switches, NFX Series Devices, and SRX Series Firewalls

IN THIS SECTION

Use of IPFIX allows you to define a flow record template suitable for IPv4 traffic or IPv6 traffic. Templates are transmitted to the collector periodically, and the collector does not affect the router configuration. You can define template refresh rate, flow active timeout and inactive timeout.

If flow records are being sent for multiple protocol families (for example, for IPv4 and IPv6), each protocol family flow has a unique Observation Domain ID. The following sections contain additional information:

Starting with Junos OS Release 17.3R1, IPFIX flow templates are supported on QFX10002 switches.

Starting with Junos OS Release 17.4R1, IPFIX flow templates are supported on QFX10008 and QFX10016 switches.

Starting with Junos OS Release 19.4R1, IPFIX flow templates are supported on SRX4100, SRX4200, SRX4600, SRX5400, SRX5600, SRX5800, vSRX Virtual Firewall, and vSRX3.0 devices.

Starting with Junos OS Release 20.1R1, IPFIX flow templates are supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Starting with Junos OS Release 20.4R1, IPFIX flow templates are supported on NFX150, NFX250 NextGen, and NFX350 devices.

> **NOTE**: (SRX Firewalls only) Inline active flow monitoring acts on a flow with respect to session. When services offload (SOF), PMI, or both are enabled for a given session, inline active flow monitoring is not supported for that session. If a session is handled by PMI or SOF, then inline active flow monitoring is disabled for that flow.

Configuring the IPFIX Template Properties

To define the IPFIX templates, include the following statements at the `[edit services flow-monitoring version-ipfix]` hierarchy level:

```
[edit services flow-monitoring version-ipfix]
template template-name {
    options-template-id
    template-id
    observation-domain-id
    flow-active-timeout seconds;
    flow-inactive-timeout seconds;
    option-refresh-rate packets packets seconds seconds;
    template-refresh-rate packets packets seconds seconds;
```

```
    (ipv4-template | ipv6-template);
}
```

The following details apply to the configuration statements:

- You assign each template a unique name by including the `template` *`template-name`* statement.
- You then specify each template for the appropriate type of traffic by including the `ipv4-template` or `ipv6-template`.
- Within the template definition, you can optionally include values for the `flow-active-timeout` and `flow-inactive-timeout` statements. These statements have specific default and range values when they are used in template definitions; the default is 60 seconds and the range is from 10 through 600 seconds.
- You can also include settings for the `option-refresh-rate` and `template-refresh-rate` statements within a template definition. For both of these properties, you can include a timer value (in seconds) or a packet count (in number of packets). For the `seconds` option, the default value is 600 and the range is from 10 through 600. For the `packets` option, the default value is 4800 and the range is from 1 through 480,000.
- To filter IPv6 traffic on a media interface, the following configuration is supported:

```
interfaces interface-name {
    unit 0 {
        family inet6 {
            sampling {
                input;
                output;
            }
        }
    }
}
```

Restrictions

The following restrictions apply to IPFIX templates:

- Outbound Routing Engine traffic is not sampled. A firewall filter is applied as output on the egress interface, which samples packets and exports the data. For transit traffic, egress sampling works correctly. For internal traffic, the next hop is installed in the Packet Forwarding Engine but sampled packets are not exported.

- Flows are created only after the route record resynchronization operation is complete, which takes 120 seconds.
- The VLAN ID field is updated when a new flow record is created and so, any change in VLAN ID after the record has been created might not be updated in the record.

Customizing Template ID, Observation Domain ID, and Source ID for IPFIX flow Templates

Starting in Junos OS Release 14.1, you can define an IPFIX flow record template suitable for IPv4 traffic, IPv6 traffic, MPLS traffic, a combination of IPv4 and MPLS traffic, or peer AS billing traffic. Templates and the fields included in the template are transmitted to the collector periodically, and the collector need not be aware of the router configuration. You can specify the unique identifier for the version 9 and IPFIX templates. The identifier of a template is locally unique within a combination of a transport session and an observation domain. Template IDs 0 through 255 are reserved for template sets, options template sets, and other sets for future use. Template IDs of data sets are numbered from 256 through 65535. Typically, this information element or field in the template is used to define the characteristics or properties of other information elements in a template. After a restart of the export process of templates is performed, you can reassign template IDs.

This functionality to configure template ID, options template ID, observation domain ID, and source ID is supported on all routers with MPCs.

The corresponding data sets and option data sets contain the value of the template IDs and options template IDs respectively in the set ID field. This method enables the collector to match a data record with a template record.

For more information about specifying the source ID, observation domain ID, template ID, and options template ID for version 9 and IPFIX flows, see "Configuring Observation Domain ID and Source ID for Version 9 and IPFIX Flows" on page 638 and "Configuring Template ID and Options Template ID for Version 9 and IPFIX Flows" on page 643.

IPFIX Templates

For information about the definitions of the fields included in IPFIX IPv4 and IPv6 templates, see "IPFIX and Version 9 Templates" on page 447.

Verification

The following show commands are supported for IPFIX:

- `show services accounting flow inline-jflow fpc-slot` *fpc-slot*
- `show services accounting errors inline-jflow fpc-slot` *fpc-slot*

- `show services accounting status inline-jflow fpc-slot` *`fpc-slot`*

Example: Configuring IPFIX Flow Templates and Flow Sampling

The following example shows an IPFIX template configuration:

```
services {
    flow-monitoring {
        version-ipfix {
            template ipv4 {
                flow-active-timeout 60;
                flow-inactive-timeout 70;
                template-refresh-rate seconds 30;
                option-refresh-rate seconds 30;
                ipv4-template;
            }
        }
    }
}
```

```
chassis;
    fpc 0 {
        sampling-instance s1;
    }
```

The following example applies the IPFIX template to enable sampling of traffic for billing:

```
forwarding-options {
    sampling {
        instance {
            s1 {
                input {
                    rate 10;
                }
                family inet {
                    output {
                        flow-server 192.0.2.2 {
                            port 2055;
                            version-ipfix {
                                template {
```

```
                                ipv4;
                            }
                        }
                    }
                    inline-jflow {
                        source-address 198.51.100.1;
                    }
                }
            }
        }
    }
}
```

Example: Configuring Inline Active Flow Monitoring Version 9 Flow Templates and Flow Sampling

The following example shows inline Active Flow Monitoring version 9 IPv4 template configuration:

```
services {
    flow-monitoring {
        version9 {
            template ipv4-v9 {
                flow-active-timeout 60;
                flow-inactive-timeout 15;
                template-refresh-rate {
                    packets 1000;
                }
                option-refresh-rate {
                    seconds 100;
                }
                ipv4-template;
            }
        }
    }
}
```

The following example shows inline Active Flow Monitoring version 9 IPv6 template configuration:

```
services {
    flow-monitoring {
        version9 {
            template ipv6-v9 {
                flow-active-timeout 60;
                flow-inactive-timeout 15;
                template-refresh-rate {
                    packets 1000;
                }
                option-refresh-rate {
                    seconds 100;
                }
                Ipv6-template;
            }
        }
    }
}
```

The following example shows inline Active Flow Monitoring version 9 IPv4 sampling traffic and export configuration:

```
forwarding-options {
    sampling {
        traceoptions {
            file testsample size 1g world-readable;
            flag all;
        }
        instance {
            sample-ins1 {
                input {
                    rate 1;
                    run-length 0;
                }
                family inet {
                    output {
                        flow-server 10.207.18.113 {
                            port 2055;
                            version9 {
                                template {
```

```
                                ipv4-v9;
                            }
                        }
                    }
                    inline-jflow {
                        source-address 10.207.18.232;
                        flow-export-rate 2;
                    }
                }
            }
        }
    }
}
```

The following example shows inline Active Flow Monitoring version 9 IPv6 sampling traffic and export configuration:

```
forwarding-options {
    sampling {
        traceoptions {
            file testsample size 1g world-readable;
            flag all;
        }
        instance {
            sample-ins1 {
                input {
                    rate 1;
                    run-length 0;
                }
                family inet {
                    output {
                        flow-server 2001::2 {
                            port 4739;
                            version9 {
                                template {
                                    ipv6-v9;
                                }
                            }
                        }
                        inline-jflow {
                            source-address 2001::1;
```

```
                            flow-export-rate 2;
                        }
                    }
                }
            }
        }
    }
}
```

The following example shows inline Active Flow Monitoring version 9 sampling interface binding (using interface):

```
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {  // 'inet6' for IPv6 protocol
            sampling {
                input;
                output;
            }
        }
    }
}
```

The following example shows inline Active Flow Monitoring version 9 sampling interface binding with firewall filter (using filters):

```
firewall {
    family  inet {   // 'inet6' for IPv6 protocol
    filter ipv4_sample {
        term default {
            then {
                accept;
                sample;
            }
        }
    }
}
```

Example: Configuring IPFIX Flow Templates and Flow Sampling

The following example shows IPFIX IPv4 template configuration:

```
flow-monitoring {
    version-ipfix {
        template ipv4-ipfix {
            flow-active-timeout 60;
            flow-inactive-timeout 60;
            template-refresh-rate {
                packets 1000;
                seconds 30;
            }
            option-refresh-rate {
                packets 500;
                seconds 60;
            }
            ipv4-template;
        }
    }
}
```

The following example shows IPFIX IPv6 template configuration:

```
flow-monitoring {
    version-ipfix {
        template ipv6-ipfix {
            flow-active-timeout 60;
            flow-inactive-timeout 60;
            template-refresh-rate {
                packets 1000;
                seconds 30;
            }
            option-refresh-rate {
                packets 500;
                seconds 60;
            }
            Ipv6-template;
        }
    }
}
```

The following example shows IPFIX IPv4 sampling traffic and export configuration:

```
forwarding-options {
    sampling {
        traceoptions {
            file testsample size 1g world-readable;
            flag all;
        }
        instance {
            sample-ins1 {
                input {
                    rate 1;
                    run-length 0;
                }
                family inet {
                    output {
                        flow-server 10.207.18.113 {
                            port 4739;
                            version-ipfix {
                                template {
                                    ipv4-ipfix;
                                }
                            }
                        }
                        inline-jflow {
                            source-address 10.207.18.232;
                            flow-export-rate 2;
                        }
                    }
                }
            }
        }
    }
}
```

The following example shows IPFIX IPv6 sampling traffic and export configuration:

```
forwarding-options {
    sampling {
        traceoptions {
            file testsample size 1g world-readable;
```

```
                flag all;
            }
            instance {
                sample-ins1 {
                    input {
                        rate 1;
                        run-length 0;
                    }
                    family inet {
                        output {
                            flow-server 2001::2 {
                                port 4739;
                                version9 {
                                    template {
                                        ipv6-ipfix;
                                    }
                                }
                            }
                            inline-jflow {
                                source-address 2001::1;
                                flow-export-rate 2;
                            }
                        }
                    }
                }
            }
        }
    }
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.4R1	Starting with Junos OS Release 20.4R1, IPFIX flow templates are supported on NFX150, NFX250 NextGen, and NFX350 devices.
20.1R1	Starting with Junos OS Release 20.1R1, IPFIX flow templates are supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

19.4R1	Starting with Junos OS Release 19.4R1, IPFIX flow templates are supported on SRX4100, SRX4200, SRX4600, SRX5400, SRX5600, SRX5800, vSRX Virtual Firewall, and vSRX3.0 devices.
17.4R1	Starting with Junos OS Release 17.4R1, IPFIX flow templates are supported on QFX10008 and QFX10016 switches.
17.2R1	Starting with Junos OS Release 17.3R1, IPFIX flow templates are supported on QFX10002 switches.
14.2	Fragment Identification (Starting in Junos OS Release 14.2)
14.2	IPv6 Extension Headers (Starting in Junos OS Release 14.2)
14.1	Starting in Junos OS Release 14.1, you can define an IPFIX flow record template suitable for IPv4 traffic, IPv6 traffic, MPLS traffic, a combination of IPv4 and MPLS traffic, or peer AS billing traffic.

RELATED DOCUMENTATION

Understanding Flow Aggregation | 594

Including Fragmentation Identifier and IPv6 Extension Header Elements in IPFIX Templates on MX Series Routers | 653

Enabling Flow Aggregation | 595

Configuring Flow Aggregation on MX, M and T Series Routers and NFX250 to Use Version 5 or Version 8 cflowd | 596

Configuring Flow Aggregation on MX, M, vMX and T Series Routers and NFX250 to Use Version 9 Flow Templates | 601

Configuring Flow Aggregation to Use IPFIX Flow Templates on PTX Series Routers

IN THIS SECTION

Use of IPFIX allows you to define a flow record template suitable for IPv4 traffic or IPv6 traffic. Templates are transmitted to the collector periodically, and the collector is not aware of the router configuration. You can define template refresh rate, flow active timeout and inactive timeout.

If flow records are being sent for multiple protocol families (for example, for IPv4 and IPv6), each protocol family flow will have a unique Observation Domain ID.

To learn about the fields included in the templates, see "Understanding Inline Active Flow Monitoring" on page 439.

Configuring the IPFIX Template Properties

To define the IPFIX templates, include the following statements at the `[edit services flow-monitoring version-ipfix]` hierarchy level:

```
[edit services flow-monitoring version-ipfix]
template name {
    options-template-id
    template-id
    observation-domain-id
    flow-active-timeout seconds;
    flow-inactive-timeout seconds;
    option-refresh-rate packets packets seconds seconds;
    template-refresh-rate packets packets seconds seconds;
    (ipv4-template | ipv6-template);
}
```

The following details apply to the configuration statements:

- You assign each template a unique name by including the `template` *name* statement.
- You then specify each template for the appropriate type of traffic by including the `ipv4-template` or `ipv6-template`.
- Within the template definition, you can optionally include values for the `flow-active-timeout` and `flow-inactive-timeout` statements. These statements have specific default and range values when they are

used in template definitions; the default is 60 seconds and the range is from 10 through 600 seconds.

- You can also include settings for the `option-refresh-rate` and `template-refresh-rate` statements within a template definition. For both of these properties, you can include a timer value (in seconds) or a packet count (in number of packets). For the `seconds` option, the default value is 600 and the range is from 10 through 600. For the `packets` option, the default value is 4800 and the range is from 1 through 480,000.

- To filter IPv6 traffic on a media interface, the following configuration is supported:

```
interfaces interface-name {
    unit 0 {
        family inet6 {
            sampling {
                input;
                output;
            }
        }
    }
}
```

Restrictions

The following restrictions apply to IPFIX templates:

- Outbound Routing Engine traffic is not sampled. A firewall filter is applied as output on the egress interface, which samples packets and exports the data. For transit traffic, egress sampling works correctly. For internal traffic, the next hop is installed in the Packet Forwarding Engine but sampled packets are not exported.

- Flows are created only after the route record resynchronization operation is complete, which takes 120 seconds.

Customizing Template ID, Observation Domain ID, and Source ID for IPFIX Flow Templates

NOTE: For PTX Series routers with third generation FPCs installed, the FPC’s slot number is used for the observation domain ID.

Use of IPFIX flow templates allow you to define a flow record template suitable for IPv4 traffic, IPv6 traffic, MPLS traffic, a combination of IPv4 and MPLS traffic, or peer AS billing traffic. Templates and the fields included in the template are transmitted to the collector periodically, and the collector does not need to be aware of the router configuration. Template IDs 0 through 255 are reserved for template sets, options template sets, and other sets for future use. Template IDs of data sets are numbered from 256 through 65535. Typically, this information element or field in the template is used to define the characteristics or properties of other information elements in a template. After a restart of the export process of templates is performed, template IDs can be reassigned.

The corresponding data sets and option data sets contain the value of the template IDs and options template IDs respectively in the set ID field. This method enables the collector to match a data record with a template record.

Verification

The following show commands are supported for IPFIX:

- `show services accounting flow inline-jflow fpc-slot` *`fpc-slot`*
- `show services accounting errors inline-jflow fpc-slot` *`fpc-slot`*
- `show services accounting status inline-jflow fpc-slot` *`fpc-slot`*

Example: Configuring an IPFIX Flow Template and Flow Sampling

The following is a sample IPFIX template configuration:

```
services {
    flow-monitoring {
        version-ipfix {
            template ipv4 {
                flow-active-timeout 60;
                flow-inactive-timeout 70;
                template-refresh-rate seconds 30;
                option-refresh-rate seconds 30;
                ipv4-template;
            }
        }
```

```
    }
}
```

```
chassis;
    fpc 0 {
        sampling-instance s1;
    }
```

The following example applies the IPFIX template to enable sampling of traffic for billing:

```
forwarding-options {
    sampling {
        instance {
            s1 {
                input {
                    rate 10;
                }
                family inet {
                    output {
                        flow-server 11.11.4.2 {
                            port 2055;
                            version-ipfix {
                                template {
                                    ipv4;
                                }
                            }
                        }
                        inline-jflow {
                            source-address 11.11.2.1;
                        }
                    }
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

Configuring Observation Domain ID and Source ID for Version 9 and IPFIX Flows

IN THIS SECTION

For IPFIX flows, an identifier of an observation domain is locally unique to an exporting process of the templates. The export process uses the observation domain ID to uniquely identify to the collection process in which the flows were metered. We recommend that you configure this ID to be unique for each IPFIX flow. A value of 0 indicates that no specific observation domain is identified by this information element. Typically, this attribute is used to limit the scope of other information elements. If the observation domain is not unique, the collector cannot uniquely identify an IPFIX device.

For version 9 flows, a 32-bit value that identifies the Exporter Observation Domain is called the source ID. NetFlow collectors use the combination of the source IP address and the source ID field to separate different export streams originating from the same exporter.

To specify the observation domain ID for IPFIX flows, include the `observation-domain-id` *`domain-id`* statement at the `[edit services flow-monitoring version-ipfix template` *`template-name`*`]` hierarchy level.

```
[edit services flow-monitoring version-ipfix]
template template-name {
    observation-domain-id domain-id;
}
```

To specify the source ID for version 9 flows, include the `source-id` *`source-id`* statement at the `[edit services flow-monitoring version9 template` *`template-name`*`]` hierarchy level.

```
[edit services flow-monitoring version9]
template template-name {
    source-id source-id;
}
```

Considerations for MX and QFX Series

If you configure the same Observation Domain ID for different template types, such as for IPv4 and IPv6, it does not impact flow monitoring because the actual or the base observation domain ID is transmitted in the flow. The actual observation domain ID is derived from the value you configure and also in conjunction with other parameters such as the slot number, lookup chip (LU) instance, Packet Forwarding Engine instance. Such a method of computation of the observation domain ID ensures that this ID is not the same for two IPFIX devices.

Until Junos OS Release 13.3, the observation domain ID is predefined and is set to a fixed value, which is derived from the combination of FPC slot, sampling protocol, PFE Instance and LU Instance fields. This derivation creates a unique observation domain per LU per family. Starting with Junos OS Release 14.1, you can configure the observation domain ID, which causes the first 8 bits of the field to be configured.

The following modifications have been made:

- FPC slots are expanded to 8 bits to enable more slots to be configured in an MX Series Virtual Chassis configuration.
- 8 bits of the configured observation domain ID are used.
- You can configure a value for the observation domain ID in the range of 0 through 255.
- The Protocol field is increased to 3 bits to provide support for additional protocols in inline flow monitoring.
- You can associate the observation domain ID with templates by using the `observation-domain-id` *`domain-id`* statement at the `[edit services flow- monitoring version-ipfix template` *`template-name`*`]` hierarchy level.

Table 117 on page 640 describes observation domain ID values for different combinations of the configured domain ID, protocol family, FPC slot, and the Packet Forwarding Engine and lookup chip instances.

Table 117: MX Series: Example of Observation Domain ID

Configured Value	Protocol Family	FPC Slot	PFE Inst	LU Inst	Observation Domain Id Conf val rsvd 1proto slot LUInst PFEInst xxxx xxxx xxxx 1xxx xxxx xxxx xxxx xxxx
None	IPv4 (0)	1	1	0	0000 0000 0000 1000 0000 0001 0000 0001 0x00080101
None	IPv6 (1)	1	1	0	0000 0000 0000 1001 0000 0001 0000 0001 0x00090101
None	VPLS (2)	1	1	0	0000 0000 0000 1010 0000 0001 0000 0001 0x000A0101
None	MPLS (3)	1	1	0	0000 0000 0000 1011 0000 0001 0000 0001 0x000B0101
4	IPv4 (0)	1	1	0	0000 0100 0000 1000 0000 0001 0000 0001 0x04080101

Table 117: MX Series: Example of Observation Domain ID *(Continued)*

Configured Value	Protocol Family	FPC Slot	PFE Inst	LU Inst	Observation Domain Id Conf val rsvd 1proto slot LUInst PFEInst xxxx xxxx xxxx 1xxx xxxx xxxx xxxx xxxx
190	IPv4 (0)	1	1	0	1101 1110 0000 1000 0000 0001 0000 0001 0xBE080101
4	IPv4 (0)	2	1	1	0000 0100 0000 1000 0000 0010 0001 0001 0x04080211
4	IPv6 (1)	1	1	0	0000 0100 0000 1001 0000 0001 0001 0000 0x04090110
190	IPv6 (1)	1	1	0	1101 1110 0000 1001 0000 0001 0001 0000 0xBE090110
4	VPLS (2)	2	2	0	0000 0100 0000 1010 0000 0010 0010 0000 0x040A0220

Table 117: MX Series: Example of Observation Domain ID *(Continued)*

Configured Value	Protocol Family	FPC Slot	PFE Inst	LU Inst	Observation Domain Id Conf val rsvd 1proto slot LUInst PFEInst xxxx xxxx xxxx 1xxx xxxx xxxx xxxx xxxx
10	IPv4 (0)	28	2	1	0000 1010 0000 1000 0001 1100 0010 0001 0x0A081C21

Considerations for PTX Series

When you configure the observation domain ID, the software attaches the ID to a particular template type.

If you configure the same observation domain ID for two different template types, such as for IPv4 and IPv6, this does not impact flow monitoring, because the configured ID is not what is being sent. The value sent in the packets is derived from that configured value and the FPC slot value. This method ensures two IPFIX devices can never have the same value of observation domain ID. As you can see in Table 118 on page 642:

- The configurable observation domain ID value is 8 bits. Therefore, the value range is 0 to 255.
- One bit is always set to 1, ensuring that the observation domain ID value sent in the packet is never 0.

Table 118: PTX Series: Format of the Observation Domain ID Value Sent in the Packet

Configured observation domain ID value (8 bits)	(15 bits set to zero)	1 bit (set to 1)	FPC slot (8 bits)

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
22.4R1-EVO	Starting in Junos OS Evolved Release 22.4R1, you can configure the observation domain ID and source ID for the PTX10001-36MR, PTX10003, PTX10004, PTX10008, and PTX10016 routers.
22.4R1	Starting in Junos OS Release 22.4R1, you can configure the observation domain ID and source ID for the PTX1000, PTX10008, and PTX10016 routers.
17.4R1	Starting with Junos OS Release 17.4R1, IPFIX flows are supported on QFX10008 and QFX10016 switches.
17.2R1	Starting with Junos OS Release 17.2R1, IPFIX flows are supported on QFX10002 switches.

RELATED DOCUMENTATION

Configuring Template ID and Options Template ID for Version 9 and IPFIX Flows | 643

Configuring Template ID and Options Template ID for Version 9 and IPFIX Flows

IN THIS SECTION

Starting with Junos OS Release 14.1, you can define the template ID for version 9 and IPFIX templates for inline flow monitoring. To specify the template ID for version 9 flows, include the `template-id` *`id`* statement at the `[edit services flow-monitoring version9 template` *`template-name`*`]` hierarchy level.

NOTE: Template ID is implemented differently on SRX Series Firewalls. You cannot configure the template ID, instead you should assign the template ID dynamically.

```
[edit services flow-monitoring version9]
template template-name {
    template-id id;
}
```

To specify the template ID for version IPFIX flows, include the `template-id` statement at the `[edit services flow-monitoring version-ipfix template template-name]` hierarchy level.

```
[edit services flow-monitoring version-ipfix]
template template-name {
    template-id id;
}
```

To specify the options template ID for version 9 flows, include the `options-template-id` statement at the `[edit services flow-monitoring version9 template template-name]` hierarchy level.

```
[edit services flow-monitoring version9]
template template-name {
    options-template-id id;
}
```

To specify the options template ID for IPFIX flows, include the `options-template-id` statement at the `[edit services flow-monitoring version-ipfix template template-name]` hierarchy level. The template ID and options template ID can be a value in the range of 1024 through 65535. For PTX Series routers, if you do not configure the template ID or options template ID, the software assigns an ID in the default range of 256-511, and the ID is different for each template.

```
[edit services flow-monitoring version-ipfix]
template template-name {
    options-template-id id;
}
```

Considerations for MX and QFX Series

The template ID and options template ID can be a value in the range of 1024 through 65535. If you do not configure values for the template ID and options template ID, default values are assumed for these IDs, which are different for the various address families. If you configure the same template ID or options template ID value for different address families, such a setting is not processed properly and might cause unexpected behavior. For example, if you configure the same template ID value for both IPv4 and IPv6, the collector validates the export data based on the template ID value that it last receives. In this case, if IPv6 is configured after IPv4, the value is effective for IPv6 and the default value is used for IPv4.

Up to Junos OS Release 13.3R1, the default values of template IDs for IPFIX flows for the different protocols or address families are:

- IPv4 IPFIX flow template ID—256
- IPv6 IPFIX flow template ID—257
- VPLS IPFIX flow template ID—258
- MPLS IPFIX flow template ID—259

Starting with Junos OS Release 14.1R1, the default values of template IDs for version 9 flows for the different protocols or address families are:

- IPv4 version 9 flow template ID—320
- IPv6 version 9 flow template ID—321
- VPLS version 9 flow template ID—322
- MPLS version 9 flow template ID—323

Up to Junos OS Release 13.3R1, the default values of options template IDs for IPFIX flows for the different protocols or address families are:

- IPv4 IPFIX flow options template ID—512
- IPv6 IPFIX flow options template ID—513
- VPLS IPFIX flow options template ID—514
- MPLS IPFIX flow options template ID—515

Starting with Junos OS Release 14.1R1, the default values of options template IDs for version 9 flows for the different protocols or address families are:

- IPv4 version 9 flow options template ID—576

- IPv6 version 9 flow options template ID—577
- VPLS version 9 flow options template ID—578
- MPLS version 9 flow options template ID—579

Table 119 on page 646 describes for the MX Series the values of data template and option template IDs for different protocols with default and configured values for IPFIX flows.

Table 119: MX Series: Values of Template and Option Template IDs for IPFIX Flows

Family	Configured Value	Data Template	Option Template
IPv4	None	256	576
IPv4	1024-65535	1024-65535	1024-65535
IPv6	None	257	577
IPv6	1024-65535	1024-65535	1024-65535
VPLS	None	258	578
VPLS	1024-65535	1024-65535	1024-65535
MPLS	None	259	579
MPLS	1024-65535	1024-65535	1024-65535

Table 120 on page 646 describes for the MX Series the values of data template and option template IDs for different protocols with default and configured values for version 9 flows.

Table 120: MX Series: Values of Template and Option Template IDs for Version 9 Flows

Family	Configured Value	Data Template	Option Template
IPv4	None	320	576

Table 120: MX Series: Values of Template and Option Template IDs for Version 9 Flows *(Continued)*

Family	Configured Value	Data Template	Option Template
IPv4	1024-65535	1024-65535	1024-65535
IPv6	None	321	577
IPv6	1024-65535	1024-65535	1024-65535
VPLS	None	322	578
VPLS	1024-65535	1024-65535	1024-65535
MPLS	None	323	579
MPLS	1024-65535	1024-65535	1024-65535

Table 121 on page 647 describes for the MX Series the values of data template and option template IDs for different protocols with default and configured values for IPFIX flows.

Table 121: MX Series: Values of Template and Option Template IDs for IPFIX Flows

Configured Value	Protocol Family	FPC Slot	PFE Inst	LU Inst	Observation Domain Id Conf val rsvd 1proto slot LUInst PFEInst xxxx xxxx xxxx 1xxx xxxx xxxx xxxx xxxx
None	IPv4 (0)	1	1	0	0000 0000 0000 1000 0000 0001 0000 0001 0x00080101

Table 121: MX Series: Values of Template and Option Template IDs for IPFIX Flows *(Continued)*

Configured Value	Protocol Family	FPC Slot	PFE Inst	LU Inst	Observation Domain Id Conf val rsvd 1proto slot LUInst PFEInst xxxx xxxx xxxx 1xxx xxxx xxxx xxxx xxxx
None	IPv6 (1)	1	1	0	0000 0000 0000 1001 0000 0001 0000 0001 0x00090101
None	VPLS (2)	1	1	0	0000 0000 0000 1010 0000 0001 0000 0001 0x000A0101
None	MPLS (3)	1	1	0	0000 0000 0000 1011 0000 0001 0000 0001 0x000B0101
4	IPv4 (0)	1	1	0	0000 0100 0000 1000 0000 0001 0000 0001 0x04080101
190	IPv4 (0)	1	1	0	1101 1110 0000 1000 0000 0001 0000 0001 0xBE080101

Table 121: MX Series: Values of Template and Option Template IDs for IPFIX Flows *(Continued)*

Configured Value	Protocol Family	FPC Slot	PFE Inst	LU Inst	Observation Domain Id Conf val rsvd 1proto slot LUInst PFEInst xxxx xxxx xxxx 1xxx xxxx xxxx xxxx xxxx
4	IPv4 (0)	2	1	1	0000 0100 0000 1000 0000 0010 0001 0001 0x04080211
4	IPv6 (1)	1	1	0	0000 0100 0000 1001 0000 0001 0001 0000 0x04090110
190	IPv6 (1)	1	1	0	1101 1110 0000 1001 0000 0001 0001 0000 0xBE090110
4	VPLS (2)	2	2	0	0000 0100 0000 1010 0000 0010 0010 0000 0x040A0220
10	IPv4 (0)	28	2	1	0000 1010 0000 1000 0001 1100 0010 0001 0x0A081C21

Considerations for PTX Series

If you choose to configure the template ID and options template ID, the range is 1024 to 65520. If you do not configure these IDs, the default values that are set are in the range 256-511 and are different for each template.

You can configure the `template-id` and `option-template-id` statements for family `inet`, `inet6`, and `mpls` only.

You must not configure the same IDs for different templates (option or data template).

> **NOTE**: The operating system does not check to ensure that you do not configure the same ID value for different templates. If you configure the same ID value, such a setting is not processed properly and might cause unexpected behavior.

The template ID or options template ID range [configured `template-id` or `options-template-id` value + 20) is reserved and you must not configure any another ID in this range. The difference between configured template IDs or options template IDs across families should be at least 20; for example, if `template-id 1056` is configured for family `inet`, you should not configure template IDs in the range of 1056 to 1075 for any other families.

For Junos OS, if you change the template ID or options template ID, all flows are inactively timed out. New flows are learned afresh.

For Junos OS Evolved, if you change the template ID or options template ID, this change does not impact the flows.

Table 122 on page 651 summarizes the data and option template ID values that correspond to configured values for template IDs, by family.

Table 122: PTX Series: Data and Option Template IDs

Template Family	Configured Value for Data or Option Template ID	Tunnel Observation Knob	Range of Template ID For Primary and Secondary Template	Template Type	Reserved Data Template ID	Reserved Option Template ID
IPv4 (Junos OS Evolved)	T1	Not specified	(T1...T1+20)	IPv4	T1	T1 (Option Template-System Scope) T1+1 (Option Template – Template Scope)
IPv4 (Junos OS)	T1	MPLS-O-UDP	(T1...T1+20)	IPv4	T1	T1 (Option Template-System Scope) T1+1 (Option Template – Template Scope)
				IPv4-MPLS-IPv4	T1+1	T1 (Option Template-System Scope) T1+1 (Option Template – Template Scope)

Table 122: PTX Series: Data and Option Template IDs *(Continued)*

Template Family	Configured Value for Data or Option Template ID	Tunnel Observation Knob	Range of Template ID For Primary and Secondary Template	Template Type	Reserved Data Template ID	Reserved Option Template ID
				IPv4-MPLS-IPv6	T1+2	T1 (Option Template-System Scope) T1+1 (Option Template – Template Scope)
IPv6	T2	Not specified	(T2...T2+20)	IPv6	T2	T2 (Option Template-System Scope) T2+1 (Option Template – Template Scope)
MPLS	T3	No	(T3...T3+20)	MPLS	T3	T3 (Option Template-System Scope) T3+1 (Option Template – Template Scope)
		IPv4	(T3+1...T3+20)	MPLS-IPv4	T3+1	
		IPv6	(T3+1...T3+20)	MPLS-IPv6	T3+2	
		MPLS-O-UDP (Junos OS only)	(T3+1...T3+20)	MPLS-IPv4+UDP +MPLS_IPv4	T3+3	

Table 122: PTX Series: Data and Option Template IDs *(Continued)*

Template Family	Configured Value for Data or Option Template ID	Tunnel Observation Knob	Range of Template ID For Primary and Secondary Template	Template Type	Reserved Data Template ID	Reserved Option Template ID
				MPLS-IPv4+UDP +MPLS_IPv6	T3+4	

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
22.4R1-EVO	Starting in Junos OS Evolved Release 22.4R1, you can configure the option template ID and template ID for the PTX10001-36MR, PTX10003, PTX10004, PTX10008, and PTX10016 routers.
22.4R1	Starting in Junos OS Release 22.4R1, you can configure the options template ID and the template ID for the PTX1000, PTX10008, and PTX10016 routers.
17.2R1	Starting with Junos OS Release 17.2R1, IPFIX templates are supported on QFX10002 switches.

RELATED DOCUMENTATION

Configuring Observation Domain ID and Source ID for Version 9 and IPFIX Flows | 638

Including Fragmentation Identifier and IPv6 Extension Header Elements in IPFIX Templates on MX Series Routers

Starting with Junos OS Release 14.2, the following attributes can be contained in IPFIX flow templates that are sent to the flow collector:

- fragmentIdentification (element ID 54)

- ipv6ExtensionHeaders (element ID 64)

A flow can receive many fragments in a given interval. For a given set of fragments of a packet, there is a unique fragment Identification. Hence, multiple such values can be received in a given interval. RFC 5102 for fragmentIdentification 54 does not clearly indicate which fragment identification needs to be shipped in the flow record information (first fragment observed after sending the flow record information or the last observed before shipping the flow record information). However, the last observed fragment Identification for a given flow is also transmitted to the flow collector.

Unlike in IPv4, IPv6 routers never fragment IPv6 packets. Packets exceeding the size of the maximum transmission unit of the destination link are dropped and this condition is signaled by a Packet Too Big ICMPv6 type 2 message to the originating node, similarly to the IPv4 method when the Don't Fragment (DF) bit is set.

The fragmentIdentification element is supported for both IPv4 and IPv6 flow templates. The fragmentIdentification element is added in the record template. The fragmentIdentification attribute is 32 bits in size for both IPv4 and IPv6. For IPv6, this field is present in fragment Extension header and Fragment Identifier is updated as 0 if there is no Fragment extension header.

Ports are a part of the key used to identify a Flow and the subsequent packets after the first fragmented packet does not have the port information. For a fragmented packet that is destined to the router, the packets that are split assume different flows (the first and the subsequent packets). Also, because the port is denoted as zeroes for fragmented packets, all the traffic destined to a particular destination from a particular source might be reported as the same flow, although no association exists between them in terms of destination ports. Fragment ID is not part of the key. Although the fragment ID attribute is unique between each source and destination, they might end up as same flows in the intermediate router.

With ports being used in the key for the flow lookup, the fragmented packets of a stream are accounted in two different flows. The first fragmented packet, which contains the port information in its packet, is part of one flow. Subsequent packets after the first fragments, which do not contain the port information, are accounted under a different flow. Because the second flow does not contain the port information to identify itself, it consolidates all the other traffic streams with same source IP and destination IP address prefixes (also includes the non-first fragmented packets sent on different ports).

Destination nodes or endpoints in IPv6 are expected to perform path MTU discovery to determine the maximum size of packets to send, and the upper-layer protocol is expected to limit the payload size. However, if the upper-layer protocol is unable to do so, the sending host can use the Fragment extension header in order to perform end-to-end fragmentation of IPv6 packets. Any data link layer conveying IPv6 data must be capable of delivering an IP packet containing 1280 bytes without the need to invoke end-to-end fragmentation at the IP layer.

The ipv6ExtensionHeaders information element is a set for 32 bit fields. Each bit in this set represents one IPv6 Extension header. An extension header bit is set if that particular extension header is observed for the flow. The bit is set to 1 if any observed packet of this Flow contains the corresponding IPv6 extension header. Otherwise, if no observed packet of this Flow contained the respective IPv6 extension header, the value of the corresponding bit is 0. The ipv6ExtensionHeaders element is added in the record template. The number of flows that are created depends on the number of IPv6 packets that include the IPv6 extender header attribute.

To enable the inclusion of element ID, 54, fragmentIdentification and element ID, 64, ipv6ExtensionHeaders in IPFIX flow templates that are exported to the flow collector, include the `ipv6-extended-attrib` statement at the `[edit chassis fpc slot-number inline- services flow-table-size]` hierarchy level. Collection of IP4 fragmentation IDs occurs automatically without having to configure this setting explicitly.

```
[edit chassis]
fpc slot-number {
    inline-services {
        flow-table-size {
            ipv6-extended-attrib;
        }
    }
}
```

Starting in Junos OS Releases 17.3R4, 17.4R3, 18.1R4, 18.2R2, 18.3R2, and 18.4R1, the values of the IPv6 options and their functions that are contained in IPv6 packets are described in Table 123 on page 655.

Table 123: Values of IPv6 Options and Extension Headers in Packets

Bit Value	IPv6 Option	Next Header Code	Description
0	DST	60	Destination option header
1	HOP	0	Hop-by-hop option header
2	Res	Not applicable	Reserved
3	UNK	Not applicable	Unknown layer 4 header (compressed, encrypted, not supported)

Table 123: Values of IPv6 Options and Extension Headers in Packets *(Continued)*

Bit Value	IPv6 Option	Next Header Code	Description
4	FRA0	44	Fragment header – first fragment
5	RH	43	Routing header
6	FRA1	44	Fragmentation header – not first fragment
7	Res	Not applicable	Reserved
8 through 11	Res	Not applicable	Reserved
12	MOB	135	IPv6 mobility (RFC3775)
13	ESP	50	Encrypted security payload
14	AH	51	Authentication header
15	PAY	108	Payload compression header
16 through 31	Res	Not applicable	Reserved

For Junos OS Releases prior to 17.3R4, 17.4R3, 18.1R4, 18.2R2, and 18.3R2, the values of the IPv6 options and their functions that are contained in IPv6 packets are described in Table 124 on page 656.

Table 124: Values of IPv6 Options and Extension Headers in Packets

Bit Value	IPv6 Option	Next Header Code	Description
0	Res	Not applicable	Reserved

Table 124: Values of IPv6 Options and Extension Headers in Packets *(Continued)*

Bit Value	IPv6 Option	Next Header Code	Description
1	FRA1	44	Fragmentation Header
2	RH	43	Routing Header
3	FRA0	44	Fragment Header – First Fragment
4	UNK	Not applicable	Unknown Layer 4 header (compressed, encrypted, not supported)
5	Res	Not applicable	Reserved
6	HOP	0	Hop-by-hop option header
7	DST	60	Destination option header
8	PAY	108	Payload compression header
9	AH	51	Authentication header
10	ESP	50	Encrypted security payload
11 through 31	Res	Not applicable	Reserved

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.3R4	Starting in Junos OS Releases 17.3R4, 17.4R3, 18.1R4, 18.2R2, 18.3R2, and 18.4R1, the values of the IPv6 options and their functions that are contained in IPv6 packets are described in Table 123 on page 655.

RELATED DOCUMENTATION

Configuring Inline Active Flow Monitoring to Use IPFIX Flow Templates on MX, vMX and T Series Routers, EX Series Switches, NFX Series Devices, and SRX Series Firewalls | 621

ipv6-extended-attrib

Directing Replicated Flows from M and T Series Routers to Multiple Flow Servers

IN THIS SECTION

You can configure replication of the sampled flow records for use by multiple flow servers. You can use either sampling based on the Routing Engine, using cflowd version 5 or version 8, or sampling based on the services PIC, using flow aggregation version 9.

Directing Replicated Routing Engine–Based Sampling Flows to Multiple Servers

Routing Engine–based sampling supports up to eight flow servers for both cflowd version 5 and version 8 configurations. The total number of servers is limited to eight regardless of how many are configured for cflowd v5 or v8.

When you configure cflowd-based sampling, the export packets are replicated to all flow servers configured to receive them. If two servers are configured to receive v5 records, both the servers receive records for a specified flow.

NOTE: With Routing Engine–based sampling, if multiple flow servers are configured with version 8 export format, all of them must use the same aggregation type. For example, all servers receiving version 8 export can be configured for `source-destination` aggregation type.

The following configuration example allows replication of export packets to two flow servers.

```
forwarding-options {
    sampling {
        instance inst1 {
            input {
                rate 1;
            }
            family inet;
                output {
                    flow-server 10.10.3.2 {
                        port 2055;
                        version 5;
                        source-address 192.168.164.119;
                    }
                    flow-server 172.17.20.62 {
                        port 2055;
                        version 5;
                        source-address 192.168.164.119;
                    }
                }
            }
        }
    }
}
```

Directing Replicated Version 9 Flow Aggregates to Multiple Servers

The export packets generated for a template are replicated to all the flow servers that are configured to receive information for that template. The maximum number of servers supported is eight.

This also implies that periodic updates required by version 9 (RFC 3954) are sent to each configured collector. The following updates are sent periodically as part of this requirement:

- Options data

- Template definition

The refresh period for options data and template definition is configured on a per-template basis at the `[edit services flow-monitoring]` hierarchy level.

The following configuration example allows replication of version 9 export packets to two flow servers.

```
forwarding-options {
    sampling {
        instance inst1 {
            input {
                rate 1;
            }
            family inet;
                output {
                    flow-server 10.10.3.2 {
                        port 2055;
                        version9 {
                            template {
                                ipv4;
                            }
                        }
                    }
                    flow-server 172.17.20.62 {
                        port 2055;
                        version9 {
                            template {
                                ipv4;
                            }
                        }
                    }
                }
                flow-inactive-timeout 30;
                flow-active-timeout 60;
                interface sp-4/0/0 {
                    source-address 10.10.3.4;
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

Active Flow Monitoring Overview | 54

Configuring Flow Monitoring | 5

Configuring Services Interface Redundancy with Flow Monitoring | 72

Example: Configure Active Monitoring on a Logical System for MX Series and PTX Series Routers | 58

Logging cflowd Flows on M and T Series Routers Before Export

To collect the cflowd flows in a log file before they are exported, include the `local-dump` statement at the `[edit forwarding-options sampling output flow-server *hostname*]` hierarchy level:

```
[edit forwarding-options sampling output flow-server hostname]
local-dump;
```

By default, the flows are collected in **/var/log/sampled**; to change the filename, include the `filename` statement at the `[edit forwarding-options sampling traceoptions]` hierarchy level. For more information about changing the filename, see "Configuring Traffic Sampling Output" on page 422.

NOTE: Because the `local-dump` statement adds extra overhead, you should use it only while debugging cflowd problems, not during normal operation.

The following is an example of the flow information. The AS number exported is the origin AS number. All flows that belong under a cflowd header are dumped, followed by the header itself:

```
Jun 27 18:35:43 v5 flow entry
Jun 27 18:35:43    Src addr: 192.0.2.1
Jun 27 18:35:43    Dst addr: 198.51.100.15
Jun 27 18:35:43    Nhop addr: 198.51.100.240
Jun 27 18:35:43    Input interface: 5
Jun 27 18:35:43    Output interface: 3
Jun 27 18:35:43    Pkts in flow: 15
Jun 27 18:35:43    Bytes in flow: 600
Jun 27 18:35:43    Start time of flow: 7230
Jun 27 18:35:43    End time of flow: 7271
Jun 27 18:35:43    Src port: 26629
Jun 27 18:35:43    Dst port: 179
```

```
Jun 27 18:35:43    TCP flags: 0x10
Jun 27 18:35:43    IP proto num: 6
Jun 27 18:35:43    TOS: 0xc0
Jun 27 18:35:43    Src AS: 7018
Jun 27 18:35:43    Dst AS: 11111
Jun 27 18:35:43    Src netmask len: 16
Jun 27 18:35:43    Dst netmask len: 0
```

[... 41 more version 5 flow entries; then the following header:]

```
Jun 27 18:35:43 cflowd header:
Jun 27 18:35:43   Num-records: 42
Jun 27 18:35:43   Version: 5
Jun 27 18:35:43   low seq num: 118
Jun 27 18:35:43   Engine id: 0
Jun 27 18:35:43   Engine type: 3
```

RELATED DOCUMENTATION

Active Flow Monitoring Overview | 54

Configuring Flow Monitoring | 5

Directing Replicated Flows from M and T Series Routers to Multiple Flow Servers | 658

Configuring Services Interface Redundancy with Flow Monitoring | 72

Example: Configure Active Monitoring on a Logical System for MX Series and PTX Series Routers | 58

Configuring Next-Hop Address Learning on MX Series and PTX Series Routers for Destinations Accessible Over Multiple Paths

You can enable learning of next-hop addresses to correctly report the next hop address, output SNMP, destination IP address, and destination IP mask values in the flow records when a destination is reachable through multiple paths. By default, this behavior of learning the next-hop addresses is disabled for inline active flow monitoring.

Starting in Junos OS Release 20.3R1, if you enable learning of next-hop addresses, certain supported devices report the packet loss priority (PLP) and the first two characters of the configured forwarding class name in the IPv4 and IPv6 IPFIX flow records. The collector uses this information to derive the

DSCP bits that the packet would contain when exiting the router. The first two letters of a configured forwarding class name must be unique. For tunnel termination, 0xFF is exported in the PLP field and NULL (0) is exported in the forwarding class name field. The mapping between the PLP exported in the record and the loss priority names is as follows:

- 0x00: Low
- 0x01: Medium-low
- 0x02: Medium-high
- 0x03: High
- 0xFF: Unknown

> You do not need to enable learning of next-hop addresses on Junos OS Evolved to report the packet loss priority and forwarding class information.

When learning next-hop addresses is disabled, data is reported as follows:

- If the destination address of the sampled IPv4 flow is reachable through multiple paths, the IPv4 next hop address and the output SNMP address are reported in the flow records as the same as the gateway address and SNMP index of the first path seen in the forwarding table.
- If the destination address of the sampled IPv6 flow is reachable through multiple paths, the IPv4 next hop address and the output SNMP address are reported as 0 in the flow records.
- If the Incoming Interface (IIF) and Outgoing Interface (OIF) are not in the same VRF, then the destination IP address, destination IP mask, IPv4 next hop address, and the output SNMP address are reported as 0 in the flow records.
- The packet loss priority and forwarding class information is not reported for devices running Junos OS.

When you enable learning of next-hop addresses, the output SNMP, destination IP address, destination IP mask values, packet loss priority, and the first two characters of the configured forwarding class name in the flow records are reported correctly when a destination is reachable through multiple paths. To enable next-hop learning, include the `nexthop-learning enable` statement at the `[edit services flow-monitoring (version-ipfix | version9) template` *`template-name`*`]` hierarchy level.

```
[edit services flow-monitoring (version-ipfix | version9) template template-name]
 set nexthop-learning enable;
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.3R1	For certain supported devices, if you enable learning of next-hop addresses, the packet loss priority (PLP) and the first two characters of the configured forwarding class name are reported in the IPv4 and IPv6 IPFIX flow records. The collector uses this information to derive the DSCP bits that the packet would contain when exiting the router. The first two letters of a configured forwarding class name must be unique.
16.1R1	You can enable learning of next-hop addresses to correctly report the next hop address, output SNMP, destination IP address, and destination IP mask values in the flow records when a destination is reachable through multiple paths.

RELATED DOCUMENTATION

nexthop-learning

5

PART

Real-Time Performance Monitoring and Video Monitoring Services

CHAPTER 15

Monitoring Traffic Using Real-Time Performance Monitoring and Two-Way Active Monitoring Protocol (TWAMP)

IN THIS CHAPTER

Understanding Using Probes for Real-Time Performance Monitoring on ACX, MX, and PTX Series Routers and EX and QFX Switches

SUMMARY

Real-time performance monitoring (RPM) enables you to configure active probes to track and monitor traffic. Probes collect packets per destination and per application, including PING Internet Control Message Protocol (ICMP) packets, User Datagram Protocol and Transmission Control Protocol (UDP/TCP) packets with user-configured ports, user-configured Differentiated Services code point (DSCP) type-of-service (ToS) packets, and Hypertext Transfer Protocol (HTTP) packets. RPM provides Management Information Base (MIB) support with extensions for RFC 2925, *Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations*. For more information about SNMP MIBs that Juniper supports, see SNMP MIB Explorer.

IN THIS SECTION

Overview

When RPM is configured on a device, the device calculates network performance based on packet response time, jitter, and packet loss. The device gathers RPM statistics by sending out probes to a specified probe target, identified by an IP address. When the target receives a probe, it generates responses that are received by the device. A test can contain multiple probes. The probe type specifies the packet and protocol contents of the probe. You can use the history of the most recent 50 probes to analyze trends in your network and predict future needs.

Use Feature Explorer: Real-time Performance Monitoring and Feature Explorer: RPM and TWAMP to confirm platform and release support.

With probes, you can monitor:

- Average round-trip time
- *Jitter* of the round-trip time—The difference between the minimum and maximum round-trip time
- Maximum round-trip time
- Minimum round-trip time
- Standard deviation of the round-trip time (Junos OS only)

One-way measurements for ICMP timestamp probes include:

- Minimum, maximum, standard deviation, and jitter measurements for egress and ingress times
- Number of probe responses received
- Number of probes sent
- Percentage of lost probes

You can set thresholds to trigger SNMP traps when the values are exceeded. You can configure the following RPM thresholds:

- Ingress/egress delay
- Jitter
- Round-trip time
- Standard deviation (Junos OS only)
- Successive lost probes
- Total lost probes (per test)

You can also configure CoS classifiers and prioritization of RPM packets over regular data packets received on an input interface with the `dscp-code-points` configuration statement.

Hardware Timestamps

To account for latency or jitter in the communication of probe messages, you can enable timestamping of the probe packets (hardware timestamps). If hardware timestamps are not configured, you are using software-based timestamps. Timestamps that are generated at the software level are less accurate than they would have been with hardware timestamps.

Use Feature Explorer: Hardware timestamping of RPM probe messages, Feature Explorer: RPM hardware timestamps with routed VLAN interfaces, and Feature Explorer: RPM and TWAMP hardware-timestamp and RTT measurement to confirm platform and release support for this feature.

NOTE: RPM hardware timestamping is supported on Junos OS only, with some restrictions:

- **ACX Series routers:** The ACX710 and ACX5448 Series routers are the only ACX routers running Junos OS that support the `hardware-timestamp` statement configuration. This support started in Junos OS Release 22.3R1.

- **EX Series switches:** EX Series switches support hardware timestamps for UDP and ICMP probes. EX Series switches do not support hardware timestamps for HTTP or TCP probes.

  On the EX4300 switch, RPM timestamping is performed in the software. The RPM probes at the requester and responder devices are timestamped in the Packet Forwarding Engine instead of the Junos OS process (rmopd) that runs on the Routing Engine. This timestamping method is referred to as pseudo-hardware timestamping.

- **QFX Series switches:** QFX Series switches do not support hardware timestamps.

You can timestamp the following RPM probes to improve the measurement of latency or jitter.

- ICMP ping
- ICMP ping timestamp
- UDP ping
- UDP ping timestamp

NOTE: icmp-ping is the default probe type on devices running Junos OS.

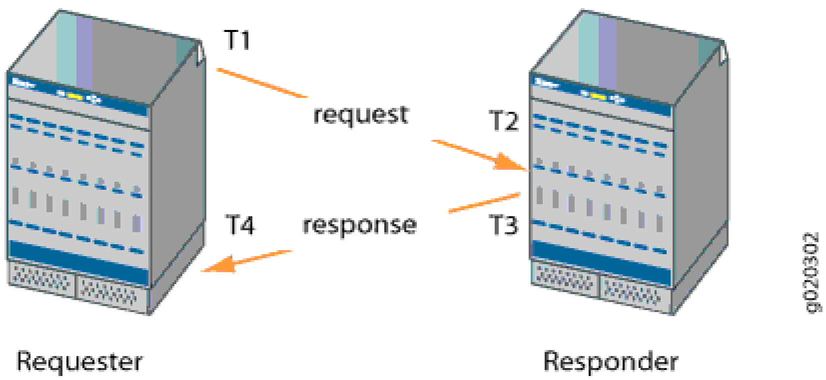
The probe packets are time-stamped with the times at which they are sent and received at both the source and destination endpoints.

You should configure the requester (the RPM client) with hardware timestamps (see Figure 62 on page 670) to get more meaningful results than you would get without the timestamps. The responder (the RPM server) does not need to be configured to support hardware timestamps. If the responder supports hardware timestamps, it timestamps the RPM probes. If the responder does not support hardware timestamps, RPM can only report round-trip measurements that include the processing time on the responder.

NOTE: On the EX4300 switch, you must configure the switch as both the requester (the RPM client) and the responder (the RPM server) to timestamp the RPM packet.

Figure 62 on page 670 shows the timestamps:

Figure 62: RPM Timestamps



- T1 is the time the packet leaves the requester port.
- T2 is the time the responder receives the packet.
- T3 is the time the responder sends the response.
- T4 is the time the requester receives the response.

The round-trip time is T4 – T1 – (T3 – T2). If the responder does not support hardware timestamps, then the round-trip time is (T4 – T1), and thus includes the processing time of the responder.

You can use RPM probes to find the following time measurements:

- Minimum round-trip time
- Maximum round-trip time
- Average round-trip time
- Standard deviation of the round-trip time
- Jitter of the round-trip time—Difference between the minimum and maximum round-trip time

The RPM feature provides a configuration option to set one-way hardware timestamps. Use one-way timestamps when you want information about one-way time, rather than round-trip times, for packets to traverse the network between the requester and the responder. As shown in Figure 62 on page 670, one-way timestamps represent the time T2 – T1 and the time from T4 – T3. Use one-way timestamps when you want to gather information about delay in each direction and to find egress and ingress jitter values.

NOTE: For correct one-way measurement, the clocks of the requester and responder must be synchronized. If the clocks are not synchronized, one-way jitter measurements and calculations can include significant variations, in some cases orders of magnitude greater than the round-trip times.

When you enable one-way timestamps in a probe, the following one-way measurements are reported:

- Minimum, maximum, standard deviation, and jitter measurements for egress and ingress times
- Number of probes sent
- Number of probe responses received
- Percentage of lost probes

Junos OS Support

Probe configuration and results

In Junos OS, probe configuration and probe results are supported by both the command-line interface (CLI) and SNMP. You set the probe options in the `test` *`test-name`* statement at the [`edit services rpm probe` *`owner`*] hierarchy level. You use the `show services rpm probe-results` command to view the results of the most recent RPM probes.

NOTE: Limitations for EX Series and QFX Series switches:

- Two-Way Active Measurement Protocol (TWAMP) is not supported on QFX switches.
- The switches do not support user-configured class-of-service (CoS) classifiers or prioritization of RPM packets over regular data packets received on an input interface.
- Timestamps:
  - If the responder does not support hardware timestamps, RPM can only report the round-trip measurements and cannot calculate round-trip jitter. (QFX Series switches do not support hardware timestamps.)
  - EX Series switches do not support hardware timestamps or pseudo-hardware timestamps for HTTP and TCP probes.
  - Timestamps apply only to IPv4 traffic.

- In-Service Software Upgrades (ISSU) and Nonstop Software Upgrades (NSSU) do not support pseudo-hardware timestamps.

To specify the packet and protocol contents of the probe, include the `probe-type` statement at the `[edit services rpm probe owner test test-name]` hierarchy level. The following probe types are supported:

- `http-get`—Sends a Hypertext Transfer Protocol (HTTP) get request to a target URL.
- `http-metadata-get`—Sends an HTTP get request for metadata to a target URL.
- `icmp-ping`—Sends ICMP echo requests to a target address.
- `icmp-ping-timestamp`—Sends ICMP timestamp requests to a target address.
- `tcp-ping`—Sends TCP packets to a target.
- `udp-ping`—Sends UDP packets to a target.
- `udp-ping-timestamp`—Sends UDP timestamp requests to a target address.

IPsec and GRE tunnel support

You can apply RPM to IPsec tunnels and GRE tunnels for PIC-based and Routing Engine-based RPM clients and servers if you are using MS-MPCs or MS-MICs. Packet Forwarding Engine-based RPM is not supported for IPsec tunnels. Support of RPM on IPSec tunnels enables service level agreement (SLA) monitoring for traffic transported in IPSec tunnels.

NOTE: RPM is not supported on logical systems.

Use Feature Explorer: RPM support for IPsec and GRE tunnels to confirm platform and release support for this feature.

RPM-tracked static routes

In Junos OS, you can also configure RPM services to determine automatically whether a path exists between a host device and its configured BGP neighbors. You can view the results of the discovery using an SNMP client. Results are stored in `pingResultsTable`, `jnxPingResultsTable`, `jnxPingProbeHistoryTable`, and `pingProbeHistoryTable`.

Use Feature Explorer: Activating or deactivating static routes on the basis of RPM test results, Feature Explorer: Tracking static RPM routes across multiple next hops, and Feature Explorer: An extension to the RPM-tracked static routes to confirm platform and release support for this feature.

For those devices that support this feature, you can use RPM probes to detect link status, and change the preferred-route state on the basis of the probe results. RPM-tracked routes can be IPv4 or IPv6, and support a single IPv4 or IPv6 next hop. You configure this feature with the `rpm-tracking` statement at the `[edit routing-options]` or `[edit routing-instances routing-options]` hierarchy level. For example, RPM probes can be sent to an IP address to determine if the link is up, and if so, the software installs a static route in the route table. RPM-tracked static routes are installed with preference 1 and thus are preferred over any existing static routes for the same prefix. For those devices that support multiple next hops, you can track up to 16 next hops for each IPv4 or IPv6 RPM-tracked static route and you can configure route preference and tag values for each IPv4 or IPv6 destination prefix.

RPM and related timestamp support on MPC, MS-MIC/MPC, and Routing Engine

Table 125 on page 673 provides information about RPM and related timestamp support on MPC, MS-MIC/MPC, and Routing Engine:

Table 125: RPM and related timestamp support for ICMP probes

Feature	Role	IP Version	Support (Y/N)	Timestamp on Routing Engine	Timestamp on MPC (hardware-timestamp)	Timestamp on MPC (si-interface)	Timestamp on MS-MIC/MPC (delegate-probes)
RPM	Client	IPv4	Y	Y (µsec) 2000 maximum probes	Y (µsec) 2000 maximum probes	N	Y (msec) 1 million maximum probes
		IPv6	Y	Y (µsec) 2000 maximum probes	N	N	Y (msec) 1 million maximum probes
	Server	IPv4	Y	Y (µsec) 2000 maximum probes	Y (µsec) 2000 maximum probes	N	Y (msec) 1 million maximum probes

Table 125: RPM and related timestamp support for ICMP probes *(Continued)*

Feature	Role	IP Version	Support (Y/N)	Timestamp on Routing Engine	Timestamp on MPC (hardware-timestamp)	Timestamp on MPC (si-interface)	Timestamp on MS-MIC/MPC (delegate-probes)
		IPv6	Y	Y (µsec) 2000 maximum probes	N	N	Y (msec) 1 million maximum probes

Junos OS Evolved Support

Probe configuration and results

Starting in Junos OS Evolved Release 20.1R1 for devices that support this feature, you can configure RPM probes. For Junos OS Evolved, RPM is configured at the `[edit services monitoring rpm]` hierarchy level. The scope of support is limited to:

- Probe generation and reception (client) as well as reflection (server) for the following RPM probe types:
  - http-get (added in Junos OS Evolved 23.4R1)

    You must set the `offload-type` statement to `none` to configure this probe type.
  - http-metadata-get (added in Junos OS Evolved 23.4R1)

    You must set the `offload-type` statement to `none` to configure this probe type.
  - icmp-ping
  - icmp-timestamp
  - tcp-ping (added in Junos OS Evolved 23.4R1)

    You must set the `offload-type` statement to `none` to configure this probe type.
  - udp-ping
  - udp-timestamp
- Probe history management

- Reporting through syslog only

Starting in Junos OS Evolved Release 21.2R1, reporting through SNMP MIB objects is supported for RPM.

Use Feature Explorer: Inline RPM Services to confirm platform and release support for Junos OS Evolved.

RPM-tracked static routes

Starting in Junos OS Evolved Release 24.4R1 for devices that support this feature, we've extended support for static route tracking to Junos OS Evolved and included Two-Way Active Measurement Protocol (TWAMP) test support as well. You use RPM or TWAMP probes to detect link status and to change the preferred-route state on the basis of the probe results. Tracked static routes can be IPv4 or IPv6, and each IPv4 and IPv6 tracked static route supports up to 16 next hops. You can also configure the metric, route preference, and tag values for each IPv4 or IPv6 destination prefix. However, you configure this feature differently on Junos OS Evolved devices; you configure the `sla-tracking` statement at the `[edit routing-options]` hierarchy level. You also use a different command, `show route sla-tracking`, to see information about these routes. For Junos OS, you would configure the `rpm-tracking` statement at the same hierarchy level and you would use the command `show route rpm-tracking` to see information about these routes.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
25.4R1	You can specify a fully qualified domain name (FQDN) as a target instead of the IPv4 or IPv6 address of the destination. The device resolves the FQDN through the Domain Name System (DNS) at the start of each test, then sends probes to the resolved address at each interval. Telemetry first displays the FQDN until resolution, then shows the destination address to help you track changes. If DNS resolution fails, the probe results retain the FQDN as the target address. If the DNS server is slow in responding, does not respond, or is not reachable, the RPM test might be slow to start or fail to start.
24.4R1-EVO	We've extended support for static route tracking to Junos OS Evolved and included Two-Way Active Measurement Protocol (TWAMP) test support as well. You use RPM or TWAMP probes to detect link status and to change the preferred-route state on the basis of the probe results. Tracked static routes can be IPv4 or IPv6, and each IPv4 and IPv6 tracked static route supports up to 16 next hops. You can also configure the metric, route preference, and tag values for each IPv4 or IPv6 destination prefix. However, you configure this feature differently on Junos OS Evolved devices; you configure the `sla-tracking` statement at the `[edit routing-options]` hierarchy level. For Junos OS, you would configure the `rpm-tracking` statement at the same hierarchy level.

23.4R1-EVO	You can configure `tcp-ping`, `http-get`, and `http-metadata-get` probes for RPM.
23.1R1-EVO	You can configure IPv6 source and target addresses for RPM probes. We've also added support for IPv6 addresses to the SNMP RFC2925a MIB control and results tables. For IPv6 RPM probes, you can enable timestamps only in the Routing Engine.
21.2R1-EVO	Reporting through SNMP MIB objects is supported for RPM.
20.4R1	For RPM-tracked static routes, you can configure route preference and tag values for each IPv4 or IPv6 destination prefix.
20.1R1-EVO	You can configure RPM probes. For Junos OS Evolved, RPM is configured at the [edit services monitoring rpm] hierarchy level.
19.3R2	RPM is not supported when you enable Next Gen Services on an MX Series router.
19.1R1	You can track up to 16 next hops for each IPv4 or IPv6 RPM-tracked static route.
19.1R1	You can enable timestamps on RPM probe messages on the Packet Forwarding Engine.
18.4R1	You can use RPM probes to detect link status, and change the preferred-route state on the basis of the probe results. RPM-tracked routes can be IPv4 or IPv6, and support a single IPv4 or IPv6 next hop. For example, RPM probes can be sent to an IP address to determine if the link is up, and if so, the software installs a static route in the route table. RPM-tracked static routes are installed with preference 1 and thus are preferred over any existing static routes for the same prefix.
17.3R1	You can apply RPM to IPsec tunnels and GRE tunnels for PIC-based and Routing Engine-based RPM clients and servers if you are using MS-MPCs or MS-MICs.

RELATED DOCUMENTATION

Configuring BGP Neighbor Discovery Through RPM | 729

Examples: Configuring BGP Neighbor Discovery on SRX Series Firewalls and MX, M, T and PTX Series Routers With RPM | 732

Configuring RPM Probes on MX Series Routers and EX Series Switches

The probe owner and test name of an RPM probe together represent a single RPM configuration instance. When you specify the test name, you also can configure the test parameters.

To configure the probe owner, test name, and test parameters, include the `probe` statement at the `[edit services rpm]` hierarchy level:

```
[edit services rpm]
probe owner {
    delegate-probes;
    test test-name {
        data-fill data;
        data-size size;
        destination-interface interface-name;
        destination-port (RPM) port;
        dscp-code-points (RPM) dscp-bits;
        hardware-timestamp;
        history-size size;
        inet6-options;
        moving-average-size number;
        one-way-hardware-timestamp;
        probe-count count;
        probe-interval seconds;
        probe-type type;
        routing-instance (RPM) instance-name;
        rpm-scale {
            destination {
                interface interface-name.logical-unit-number;
                subunit-cnt subunit-cnt;
            }
            source {
                address-base ipv4-address-base;
                count ipv4-count;
                step ipv4-step;
            }
            source-inet6 {
                address-base ipv6-address-base;
                count ipv6-count;
                step ipv6-step;
            }
            target {
```

```
                address-base ipv4-address-base;
                count ipv4-count;
                step ipv4-step;
            }
            target-inet6 {
                address-base ipv6-address-base;
                count ipv6-count;
                step ipv6-step;
            }
            tests-count tests-count;
        }
        source-address address;
        target (url url | address address);
        test-interval interval;
        thresholds (Junos OS) thresholds;
        traps traps;
        ttl [hop-count]
    }
}
```

Keep the following points in mind when you configure RPM clients and RPM servers:

- RPM is not supported on logical systems.
- PIC-based and Routing Engine-based RPM is supported for IPsec tunnels and GRE tunnels if you are using MS-MPCs or MS-MICs. Packet Forwarding Engine-based RPM is not supported for IPsec tunnels. Support of RPM on IPSec tunnels enables service level agreement (SLA) monitoring for traffic transported in IPSec tunnels.
- You can configure the generation of IPv4 `icmp-ping` and `icmp-ping-timestamp` RPM probes on an MS-MPC or MS-MIC, which increases the number of probes generated up to 1 million per second on every service-NPU compared to the number of probes that are generated on the Packet Forwarding Engine. You can configure the generation of `icmp6-ping` RPM probes on an MS-MPC or MS-MIC. To configure the generation of RPM probes on an MS-MPC or MS-MIC:
  - Include the `destination-interface` *`interface-name.logical-unit-number`* at the `[edit services rpm probe` *`owner`* `test` *`test-name`*`]` hierarchy level, and include the `delegate-probes` statement at the `[edit services rpm probe` *`owner`*`]` hierarchy level. The *interface-name.logical-unit-number* specifies a logical interface on an MS-MPC or MS-MIC slot, PIC, and port that has a valid IP address defined on it (for example, ms-1/2/1.1). The interface cannot be an aggregated multiservices interface (ams-).
  - Include the `rpm client-delegate-probes` and the `family (inet | inet6) address` *`address`* statements at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level. The *interface-name* and the

logical-unit-number must match the *interface-name.logical-unit-number* that you used for the `destination-interface`.

For RPM probes configured on an MS-MPC or MS-MIC, you cannot configure the `routing-instance` statement at the `[edit services rpm probe owner test test-name]` hierarchy level, and you cannot configure both IPv4 and IPv6 probes within the same test.

You can use additional filters to limit the output of the *show services rpm probe-results* and *show services rpm history-results* commands for RPM probes generated on an MS-MPC or MS-MIC.

- You can optimize the CLI configuration for RPM tests for IPv4. Starting in Junos OS Release 18.2R1, you can also optimize the CLI configuration for RPM tests for IPv6. This optimization allows the use of minimal RPM configuration statements to generate multiple tests (up to 100K tests) with pre-defined, reserved RPM test names. This optimization can be configured for tests with probes that are generated by either the Packet Forwarding Engine or by an MS-MPC or MS-MIC. Tests are generated for multiple combinations of source and target addresses, which are incremented based on your configuration.

  The maximum number of concurrent RPM probes supported for various Junos releases are as follows:

  - Without configuring the `delegate-probes` statement—2000 for ICMP and ICMP-timestamp probe types. For probes of other types (UDP and TCP) the limit is 500.
  - With configuring the *delegate-probes* statement)—1 Million per Service-NPU.

  > **NOTE**: One MS-MIC contains one service-NPU and one MS-MPC contains four service-NPUs.
  >
  > With configuring the *delegate-probes* statement, the RPM probes are compliant to RFC792 and RFC4443. Hence, they can be used to monitor any IP device compliant to either RFC and are able to respond to icmp-timestamp and/or icmp6-ping packets.

  Tests are first generated for all the source addresses with the initial target address, then tests are generated for all the source addresses with the next available target address, and so on. You can also configure a group that contains global values for a particular probe owner, and apply the group to the probe owner.

  To generate multiple RPM tests, configure the following:

```
[edit services rpm probe owner]
apply-groups group-name;
test test-name {
```

```
    rpm-scale {
        destination {
            interface interface-name.logical-unit-number;
            subunit-cnt subunit-cnt;
        }
        source {
            address-base ipv4-address-base;
            count ipv4-count;
            step ipv4-step;
        }
        source-inet6 {
            address-base ipv6-address-base;
            count ipv6-count;
            step ipv6-step;
        }
        target {
            address-base ipv4-address-base;
            count ipv4-count;
            step ipv4-step;
        }
        target-inet6 {
            address-base ipv6-address-base;
            count ipv6-count;
            step ipv6-step;
        }
        tests-count tests-count;
    }
}
```

The options are:

ipv4-address-base	The IPv4 source or target address that is incremented to generate the addresses used in the RPM tests.
ipv6-address-base	The IPv6 source or target address that is incremented to generate the addresses used in the RPM tests.
ipv4-step	The amount to increment the IPv4 source or target address for each generated RPM test.
ipv6-step	The amount to increment the IPv6 source or target address for each generated RPM test.

ipv4-count	The maximum number of IPv4 source or target addresses to use for the generated RPM tests.
ipv6-count	The maximum number of IPv6 source or target addresses to use for the generated RPM tests.
interface-name.logical-unit-number	The services interface that is generating RPM probes and the logical unit number that is used for the first test that is generated.
subunit-cnt	The maximum number of logical units used by the services interface in the generated tests. The first generated test uses the logical unit specified in the *interface-name.logical-unit-number* option, and each successive test increments the logical unit number by one. Once the maximum number of logical units has been used, the next generated test cycles back to the logical unit that was used in the first test.
tests-count	The maximum number of RPM tests to generate. This number must be less than or equal to the number of generated source addresses multiplied by the number of generated target addresses.

To configure a group with global values for a particular probe owner:

```
[edit groups group-name]
services {
    rpm {
        probe <*> {
            test {
                data-fill data;
                data-size size;
                dscp-code-points (RPM) dscp-bits;
                history-size size;
                moving-average-size number;
                probe-count count;
                probe-type type;
                test-interval interval;
                thresholds (Junos OS) thresholds;
            }
        }
    }
}
```

- To specify a probe owner, include the `probe` statement at the `[edit services rpm]` hierarchy level. The probe owner identifier can be up to 32 characters in length.

- To specify a test name, include the `test` statement at the `[edit services rpm probe owner]` hierarchy level. The test name identifier can be up to 32 characters in length. A test represents the range of probes over which the standard deviation, average, and jitter are calculated.

- To specify the contents of the data portion of Internet Control Message Protocol (ICMP) probes, include the `data-fill` statement at the `[edit services rpm probe owner]` hierarchy level. The value can be a hexadecimal value. The `data-fill` statement is not valid with the `http-get` or `http-metadata-get` probe types.

- To specify the size of the data portion of ICMP probes, include the `data-size` statement at the `[edit services rpm probe owner]` hierarchy level. The size can be from `0` through `65400` and the default size is `0`. The `data-size` statement is not valid with the `http-get` or `http-metadata-get` probe types.

  NOTE: If you configure the hardware timestamp feature (see "Configuring RPM Timestamping on MX, M, T, and PTX Series Routers and EX Series Switches" on page 718):

  - The `data-size` default value is 32 bytes and 32 is the minimum value for explicit configuration. The UDP timestamp probe type is an exception; it requires a minimum data size of 44 bytes.

  - The configured value on the `data-size` statement must be at least 100 bytes smaller than the default MTU of the interface of the RPM client interface.

- On MX Series routers, you configure the `destination-interface` statement to enable hardware timestamping of RPM probe packets. You specify an sp- interface to have the AS or Multiservices PIC add the hardware timestamps; for more information, see "Configuring RPM Timestamping on MX, M, T, and PTX Series Routers and EX Series Switches" on page 718. You can also include the `one-way-hardware-timestamp` statement to enable one-way delay and jitter measurements.

- To specify the User Datagram Protocol (UDP) port or Transmission Control Protocol (TCP) port to which the probe is sent, include the `destination-port` statement at the `[edit services rpm probe owner test test-name]` hierarchy level. The `destination-port` statement is used only for the UDP and TCP probe types. The value can be `7` or from `49160` through `65535`.

  When you configure either `probe-type udp-ping` or `probe-type udp-ping-timestamp` along with hardware timestamping, the value for the `destination-port` can be only 7. A constraint check prevents you from configuring any other value for the destination port in this case. This constraint does not apply when you are using one-way hardware timestamping.

- To specify the value of the Differentiated Services (DiffServ) field within the IP header, include the `dscp-code-point` statement at the `[edit services rpm probe *owner* test *test-name*]` hierarchy level. The DiffServ code point (DSCP) bits value can be set to a valid 6-bit pattern; for example, `001111`. It also can be set using an alias configured at the `[edit class-of-service code-point-aliases dscp]` hierarchy level. The default is `000000`.

- To specify the number of stored history entries, include the `history-size` statement at the `[edit services rpm probe *owner* test *test-name*]` hierarchy level. Specify a value from `0` to `512`. The default is `50`.

- To specify a number of samples for making statistical calculations, include the `moving-average-size` statement at the `[edit services rpm probe *owner* test *test-name*]` hierarchy level. Specify a value from `0` through `255`.

- To specify the number of probes within a test, include the `probe-count` statement at the `[edit services rpm probe *owner* test *test-name*]` hierarchy level. Specify a value from `1` through `15`.

- To specify the time to wait between sending packets, include the `probe-interval` statement at the `[edit services rpm probe *owner* test *test-name*]` hierarchy level. Specify a value from `1` through `255` seconds.

- To specify the packet and protocol contents of the probe, include the `probe-type` statement at the `[edit services rpm probe *owner* test *test-name*]` hierarchy level. The following probe types are supported:

  - `http-get`—Sends a Hypertext Transfer Protocol (HTTP) get request to a target URL.
  - `http-metadata-get`—Sends an HTTP get request for metadata to a target URL.
  - `icmp-ping`—Sends ICMP echo requests to a target address.
  - `icmp-ping-timestamp`—Sends ICMP timestamp requests to a target address.
  - `tcp-ping`—Sends TCP packets to a target.
  - `udp-ping`—Sends UDP packets to a target.
  - `udp-ping-timestamp`—Sends UDP timestamp requests to a target address.

  The following probe types support hardware timestamping of probe packets: `icmp-ping`, `icmp-ping-timestamp`, `udp-ping`, `udp-ping-timestamp`. The delegate probes are distributed evenly across the interval of 3 seconds to avoid the packet bursts in the network due to real-time performance monitoring (RPM). RPM syslogs are processed with the increase in the ramp up time of RPM delegates tests to 60 seconds. With RPM syslogs processed, the chances of multiple tests starting and ending at the same time are smaller, thus a potential restriction in `event-processing`.

  NOTE: Some probe types require additional parameters to be configured. For example, when you specify the `tcp-ping` or `udp-ping` option, you must configure the destination port using the `destination-port` statement. The `udp-ping-timestamp` option requires a minimum data size of 12; any smaller data size results in a commit error. The minimum data size for TCP probe packets is 1.

When you configure either `probe-type udp-ping` or `probe-type udp-ping-timestamp` along with the `one-way-hardware-timestamp` command, the value for the `destination-port` can be only 7. A constraint check prevents you for configuring any other value for the destination port in this case.

- To specify the routing instance used by ICMP probes, include the `routing-instance` statement at the `[edit services rpm probe` *owner* `test` *test-name*`]` hierarchy level. The default routing instance is Internet routing table `inet.0`.
- To specify the source IP address used for ICMP probes, include the `source-address` statement at the `[edit services rpm probe` *owner* `test` *test-name*`]` hierarchy level. If the source IP address is not one of the router's assigned addresses, the packet uses the outgoing interface's address as its source.
- To specify the source IPv6 address to be used for RPM probes that are sent from the RPM client (the device that originates the RPM packets) to the RPM server (the device that receives the RPM probes), include the `inet6-options source-address` *ipv6-address* `statement` at the `[edit services rpm probe` *owner* `test` *test-name*`]` hierarchy level. If the source IPv6 address is not one of the router's or switch's assigned addresses, the packet use the outgoing interface's address as its source.
- To specify the destination address used for the probes, include the `target` statement at the `[edit services rpm probe` *owner* `test` *test-name*`]` hierarchy level.
  - For HTTP probe types, specify a fully formed URL that includes `http://` in the URL address.
  - For all other probe types, specify an IP version 4 (IPv4) or IP version 6 (IPv6) address for the target host.
- To specify the time to wait between tests, include the `test-interval` statement at the `[edit services rpm probe` *owner* `test` *test-name*`]` hierarchy level. Specify a value from `0` through `86400` seconds. A value of 0 seconds causes the RPM test to stop after one iteration. The default value is 1.
- To specify thresholds used for the probes, include the `thresholds` statement at the `[edit services rpm probe` *owner* `test` *test-name*`]` hierarchy level. A system log message is generated when the configured threshold is exceeded. Likewise, an SNMP trap (if configured) is generated when a threshold is exceeded. The following options are supported:
  - `egress-time`—Measures maximum source-to-destination time per probe.
  - `ingress-time`—Measures maximum destination-to-source time per probe.
  - `jitter-egress`—Measures maximum source-to-destination jitter per test.
  - `jitter-ingress`—Measures maximum destination-to-source jitter per test.

- `jitter-rtt`—Measures maximum jitter per test, from 0 through 60000000 microseconds.
- `rtt`—Measures maximum round-trip time per probe, in microseconds.
- `std-dev-egress`—Measures maximum source-to-destination standard deviation per test.
- `std-dev-ingress`—Measures maximum destination-to-source standard deviation per test.
- `std-dev-rtt`—Measures maximum standard deviation per test, in microseconds.
- `successive-loss`—Measures successive probe loss count, indicating probe failure.
- `total-loss`—Measures total probe loss count indicating test failure, from 0 through 15. The default for this threshold is 1.

- Traps are sent if the configured threshold is met or exceeded. To set the trap bit to generate traps, include the `traps` statement at the `[edit services rpm probe` *`owner`* `test` *`test-name`*`]` hierarchy level. The following options are supported:
  - `egress-jitter-exceeded`—Generates traps when the jitter in egress time threshold is met or exceeded.
  - `egress-std-dev-exceeded`—Generates traps when the egress time standard deviation threshold is met or exceeded.
  - `egress-time-exceeded`—Generates traps when the maximum egress time threshold is met or exceeded.
  - `ingress-jitter-exceeded`—Generates traps when the jitter in ingress time threshold is met or exceeded.
  - `ingress-std-dev-exceeded`—Generates traps when the ingress time standard deviation threshold is met or exceeded.
  - `ingress-time-exceeded`—Generates traps when the maximum ingress time threshold is met or exceeded.
  - `jitter-exceeded`—Generates traps when the jitter in round-trip time threshold is met or exceeded.
  - `probe-failure`—Generates traps for successive probe loss thresholds crossed.
  - `rtt-exceeded`—Generates traps when the maximum round-trip time threshold is met or exceeded.
  - `std-dev-exceeded`—Generates traps when the round-trip time standard deviation threshold is met or exceeded.
  - `test-completion`—Generates traps when a test is completed.
  - `test-failure`—Generates traps when the total probe loss threshold is met or exceeded.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
18.2R1	You can also optimize the CLI configuration for RPM tests for IPv6.
18.1R1	You can configure the generation of `icmp6-ping` RPM probes on an MS-MPC or MS-MIC.
18.1R1	You can use additional filters to limit the output of the *show services rpm probe-results* and *show services rpm history-results* commands for RPM probes generated on an MS-MPC or MS-MIC.
17.4R1	You can optimize the CLI configuration for RPM tests for IPv4.
17.3R3	The delegate probes are distributed evenly across the interval of 3 seconds to avoid the packet bursts in the network due to real-time performance monitoring (RPM). RPM syslogs are processed with the increase in the ramp up time of RPM delegates tests to 60 seconds. With RPM syslogs processed, the chances of multiple tests starting and ending at the same time are smaller, thus a potential restriction in `event-processing`.
17.3R1	PIC-based and Routing Engine-based RPM is supported for IPsec tunnels and GRE tunnels if you are using MS-MPCs or MS-MICs.
17.3R1	You can configure the generation of IPv4 `icmp-ping` and `icmp-ping-timestamp` RPM probes on an MS-MPC or MS-MIC, which increases the number of probes generated up to 1 million per second on every service-NPU compared to the number of probes that are generated on the Packet Forwarding Engine.
16.1	To specify the source IPv6 address to be used for RPM probes that are sent from the RPM client (the device that originates the RPM packets) to the RPM server (the device that receives the RPM probes), include the `inet6-options source-address` *ipv6-address* `statement` at the `[edit services rpm probe` *owner* `test` *test-name*`]` hierarchy level.
16.1	For all other probe types, specify an IP version 4 (IPv4) or IP version 6 (IPv6) address for the target host.

RELATED DOCUMENTATION

Understanding Using Probes for Real-Time Performance Monitoring on ACX, MX, and PTX Series Routers and EX and QFX Switches | 667

Examples: Configuring Real-Time Performance Monitoring on MX, M, T and PTX Series Routers | 738

Real-Time Performance Monitoring for SRX Devices

SUMMARY

This section describes the real-time performance monitoring (RPM) feature that allows network operators and their customers to accurately measure the performance of the network between two endpoints.

IN THIS SECTION

RPM Overview (SRX)

IN THIS SECTION

With the RPM tool, you configure and send probes to a specified target and monitor the analyzed results to determine packet loss, round-trip time, and *jitter*.

RPM 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 Hypertext Transfer Protocol (HTTP) GET requests, Internet Control Message Protocol (ICMP) requests, and TCP and UDP requests, depending on the configuration.

This section contains the following topics:

RPM Probes

You 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 device. By analyzing the transit times to and from the remote server, the device can determine network performance statistics.

The device sends out the following probe types:

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

UDP and TCP probe types require that the remote server be configured as an RPM receiver so that it generates responses to the probes.

The RPM probe results are also available in the form of MIB objects through the SNMP protocol.

NOTE: On SRX300, SRX320, SRX340, SRX1500, SRX4600 devices and vSRX Virtual Firewall instances, when you configure basic RPM probes, the following combination of the configuration parameters is not supported:
Source address and destination port and next-hop.

Configuring RPM probe with these parameters prevents sending out RPM probes to a specified probe target. We recommend you to configure either the source address or destination port and next-hop to configure RPM probe.

RPM Tests

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.

Probe and Test Intervals

Within a test, RPM probes are sent at regular intervals, 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 the 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.

> **NOTE**: On SRX340 devices, the RPM server operation with icmp is not supported. The RPM server works fine with TCP and UDP.

Jitter Measurement with Hardware Timestamping

Jitter is the difference in relative transit time between two consecutive probes.

You can timestamp the following RPM probes to improve the measurement of latency or jitter:

- ICMP ping
- ICMP ping timestamp
- UDP ping
- UDP ping timestamp

> **NOTE**: The device supports hardware timestamping of UDP ping and UDP ping timestamp RPM probes only if the destination port is UDP-ECHO (port 7).

Timestamping takes place during the forwarding process of the device originating the probe (the RPM client), but not on the remote device that is the target of the probe (the RPM server).

The supported encapsulations on a device for timestamping are Ethernet including VLAN, synchronous PPP, and Frame Relay. The only *logical interface* supported is an **lt** services interface.

RPM probe generation with hardware timestamp can be retrieved through the SNMP protocol.

RPM Statistics

At the end of each test, the device collects the statistics for packet round-trip time, packet inbound and outbound times (for ICMP timestamp probes only), and probe loss as shown in Table 126 on page 690.

Table 126: RPM Statistics

RPM Statistics	Description
Round-Trip Times	
Minimum round-trip time	Shortest round-trip time from the Juniper Networks device to the remote server, as measured over the course of the test
Maximum round-trip time	Longest round-trip time from the Juniper Networks device to the remote server, as measured over the course of the test
Average round-trip time	Average round-trip time from the Juniper Networks device to the remote server, as measured over the course of the test
Standard deviation round-trip time	Standard deviation of the round-trip times from the Juniper Networks device to the remote server, as measured over the course of the test
Jitter	Difference between the maximum and minimum round-trip times, as measured over the course of the test
Inbound and Outbound Times (ICMP Timestamp Probes Only)	
Minimum egress time	Shortest one-way time from the Juniper Networks device to the remote server, as measured over the course of the test
Maximum ingress time	Shortest one-way time from the remote server to the Juniper Networks device, as measured over the course of the test
Average egress time	Average one-way time from the Juniper Networks device to the remote server, as measured over the course of the test
Average ingress time	Average one-way time from the remote server to the Juniper Networks device, as measured over the course of the test

Table 126: RPM Statistics *(Continued)*

RPM Statistics	Description
Standard deviation egress time	Standard deviation of the one-way times from the Juniper Networks device to the remote server, as measured over the course of the test
Standard deviation ingress time	Standard deviation of the one-way times from the remote server to the Juniper Networks device, as measured over the course of the test
Egress jitter	Difference between the maximum and minimum outbound times, as measured over the course of the test
Ingress jitter	Difference between the maximum and minimum inbound times, as measured over the course of the test
Probe Counts	
Probes sent	Total number of probes sent over the course of the test
Probe responses received	Total number of probe responses received over the course of the test
Loss percentage	Percentage of probes sent for which a response was not received

RPM Thresholds and Traps

You can configure RPM threshold values for the round-trip times, ingress (inbound) times, and egress (outbound) times that are measured for each probe, as well as for the standard deviation and jitter values that are measured for each test. Additionally, you can configure threshold values for the number of successive lost probes within a test and the total number of lost probes within a test.

If the result of a probe or test exceeds any threshold, the device generates a system log message and sends any Simple Network Management Protocol (SNMP) notifications (traps) that you have configured.

Starting in Junos OS Release 18.4R1, if the result of a probe or test exceeds the packet loss threshold, the real-time performance monitoring (RPM) test probe is marked as failed. The test probe also fails when the round-trip time (RTT) exceeds the configured threshold value. As a result, the device generates an SNMP notification (trap) and marks the RPM test as failed. RPM 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.

RPM for BGP Monitoring

When managing peering networks that are connected using Border Gateway Protocol (BGP), you might need to find out if a path exists between the Juniper Networks device and its configured BGP neighbors. You can ping each BGP neighbor manually to determine the connection status, but this method is not practical when the device has a large number of BGP neighbors configured.

In the device, you can configure RPM probes to monitor the BGP neighbors and determine if they are active.

Guidelines for Configuring RPM Probes for IPv6 (SRX Series Firewalls)

Starting with Junos OS Release 15.1X49-D10, you can configure RPM Probes for IPv6.

Keep the following guidelines in mind when you configure IPv6 addresses for RPM destinations or servers:

- IPv6 RPM uses ICMPv6 probe requests. You cannot configure ICMP or ICMP timestamp probe types.
- Only Routing Engine-based RPM is supported for IPv6 targets including VRF support, specification of the size of the data portion of ICMPv6 probes, data pattern, and traffic class.
- You can configure probes with a combination of IPv4 and IPv6 tests. However, an individual test must be either IPv4 or IPv6.
- Routing Engine-based RPM does not support hardware-based, or one-way hardware-based timestamping.
- We recommend that you include the `probe-limit` statement at the `[edit services rpm]` hierarchy level to set the limit on concurrent probes to 10. Higher concurrent probes can result in higher spikes.
- SNMP set operation is permitted only on ICMP probes and it is not supported for other probe types.
- The following table describes the IPv6 special address prefixes that you cannot configure in a probe.

IPV6 Address Type	IPV6 Address Prefix
Node-Scoped Unicast	::1/128 is the loopback address ::/128 is the unspecified address

(Continued)

IPV6 Address Type	IPV6 Address Prefix
IPv4-Mapped Addresses	::FFFF:0:0/96
IPv4-Compatible Addresses	:<ipv4-address>/96
Link-Scoped Unicast	fe80::/10
Unique-Local	fc00::/7
Documentation Prefix	2001:db8::/32
6to4	2002::/16
6bone	5f00::/8
ORCHID	2001:10::/28
Teredo	2001::/32
Default Route	::/0
Multicast	ff00::/8

- In Routing Engine-based RPM, route-trip time (RTT) spikes might occur because of queuing delays, even with a single test.
- Since RPM might open TCP and UDP ports to communicate between the RPM server and RPM client, we recommend that you use firewalls and distributed denial-of-service (DDoS) attack filters to protect against security threats.

IPv6 RPM Probes (vSRX Virtual Firewall)

Starting with Junos OS Release 15.1X49-D10, Route Engine-based RPM can send and receive IPv6 probe packets to monitor performance on IPv6 networks.

A probe request is a standard IPv6 packet with corresponding TCP, UDP, and ICMPv6 headers. A probe response is also a standard IPv6 packet with corresponding TCP, UDP, and ICMPv6 headers. No RPM header is appended to the standard packet for RE-based RPM. An IPv6-based RPM test occurs between an IPv6 RPM client and IPv6 RPM server.

NOTE: You can have both IPv4 tests and IPv6 tests in the same probe.

Configuring IPv6 RPM Probes (vSRX Virtual Firewall)

Starting with Junos OS Release 15.1X49-D10, you can configure IPv6 destination addresses for an IPv6-based RPM probe test.

To configure an IPv6 RPM test:

1. Specify the RPM probe owner for the probe you want to configure as an IPv6 test.

```
[edit services rpm]
user@host# edit probe customerA
```

2. Specify a name for the test.

```
[edit services rpm probe customerA]
user@host# edit test ipv6-test
```

3. Specify the probe type.

```
[edit services rpm probe customerA test ipv6-test]
user@host# set probe-type icmp6-ping
```

4. Specify the target address for the test.

```
[edit services rpm probe customerA test ipv6-test]
user@host# set target inet6-address 2001::2
```

5. Configure the remaining RPM test parameters.

Tuning RPM Probes (SRX Series Firewalls)

After configuring an RPM probe, you can set parameters to control probe functions, such as the interval between probes, the total number of concurrent probes that a system can handle, and the source address used for each probe packet.

To tune RPM probes:

1. Set the maximum number of concurrent probes allowed on the system to `10`.

```
[edit services rpm]
user@host# set probe-limit 10
```

2. Access the ICMP probe of customer A.

```
[edit]
user@host# edit services rpm probe customerA test icmp-test
```

3. Set the time between probe transmissions to 15 seconds.

```
[edit services rpm probe customerA test icmp-test]
user@host# set probe-interval 15
```

4. Set the number of probes within a test to `10`.

```
[edit services rpm probe customerA test icmp-test]
user@host# set probe-count 10
```

5. Set the source address for each probe packet to `192.168.2.9`. If you do not explicitly configure a source address, the address on the outgoing interface through which the probe is sent is used as the source address.

```
[edit services rpm probe customerA test icmp-test]
user@host# set source-address 192.168.2.9
```

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

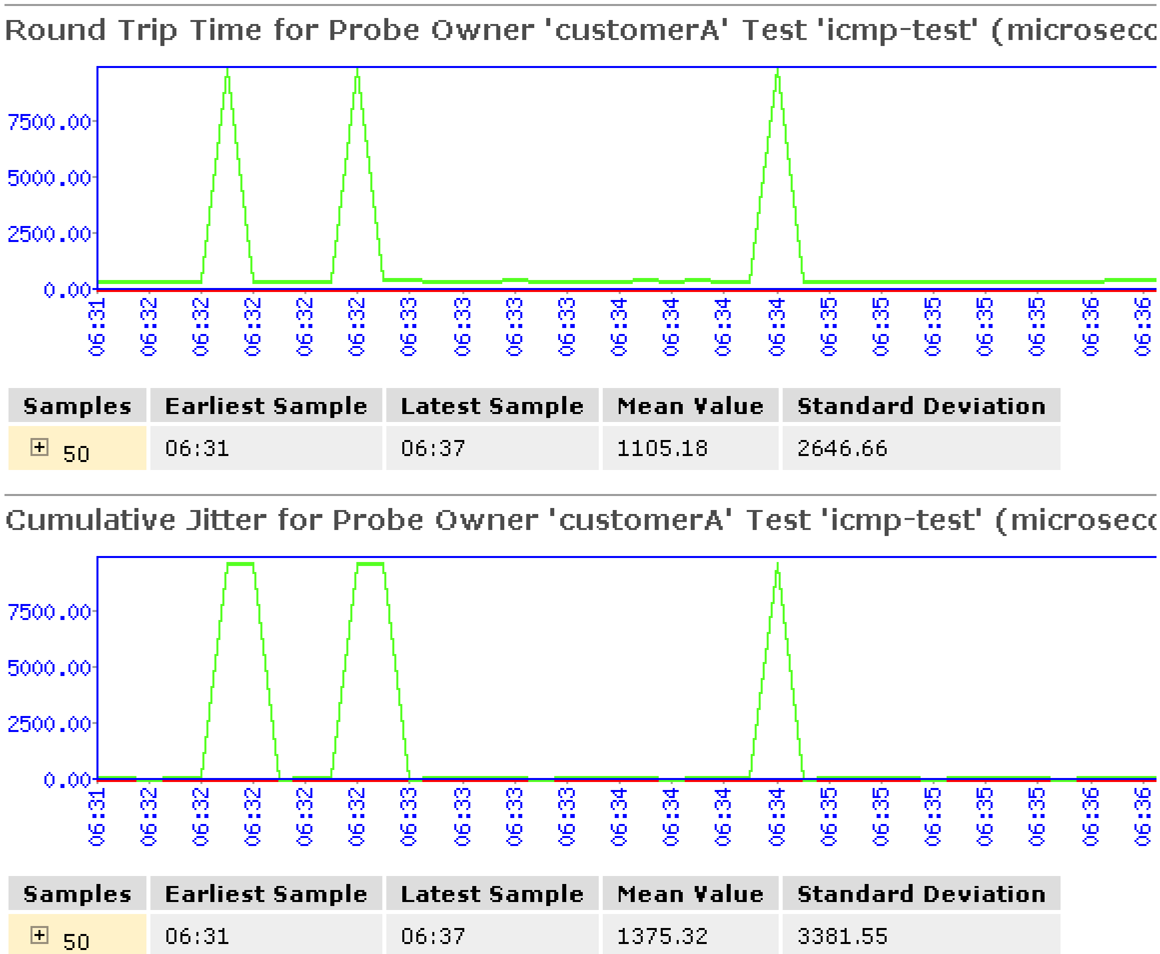
Monitoring RPM Probes (SRX Series Firewalls)

The RPM information includes the round-trip time, jitter, and standard deviation values for each configured RPM test on the device. To view these RPM properties, select **Troubleshoot>RPM>View RPM** in the J-Web user interface, or in configuration mode enter the `show` command:

```
[edit]
user@host# run show services rpm probe-results
```

In addition to the RPM statistics for each RPM test, the J-Web user interface displays the round-trip times and cumulative jitter graphically. Figure 63 on page 696 shows sample graphs for an RPM test.

Figure 63: Sample RPM Graphs



In Figure 63 on page 696, the round-trip time and jitter values are plotted as a function of the system time. Large spikes in round-trip time or jitter indicate a slower outbound (egress) or inbound (ingress) time for the probe sent at that particular time.

Table 127 on page 696 summarizes key output fields in RPM displays.

Table 127: Summary of Key RPM Output Fields

Field	Values	Additional Information
Currently Running Tests		

Table 127: Summary of Key RPM Output Fields *(Continued)*

Field	Values	Additional Information
Graph		Click the **Graph** link to display the graph (if it is not already displayed) or to update the graph for a particular test.
Owner	Configured owner name of the RPM test.	–
Test Name	Configured name of the RPM test.	–
Probe Type	Type of RPM probe configured for the specified test: • **http-get** • **http-get-metadata** • **icmp-ping** • **icmp6-ping** • **icmp-ping-timestamp** • **tcp-ping** • **udp-ping**	–
Target Address	IPv4 address, IPv6 address, or URL of the remote server that is being probed by the RPM test.	–
Source Address	Explicitly configured IPv4 or IPv6 source address that is included in the probe packet headers.	If no source address is configured, the RPM probe packets use the outgoing interface as the source address, and the Source Address field is empty.
Minimum RTT	Shortest round-trip time from the Juniper Networks device to the remote server, as measured over the course of the test.	–

Table 127: Summary of Key RPM Output Fields *(Continued)*

Field	Values	Additional Information
Maximum RTT	Longest round-trip time from the Juniper Networks device to the remote server, as measured over the course of the test.	–
Average RTT	Average round-trip time from the Juniper Networks device to the remote server, as measured over the course of the test.	–
Standard Deviation RTT	Standard deviation of round-trip times from the Juniper Networks device to the remote server, as measured over the course of the test.	–
Probes Sent	Total number of probes sent over the course of the test.	–
Loss Percentage	Percentage of probes sent for which a response was not received.	–
Round-Trip Time for a Probe		
Samples	Total number of probes used for the data set.	The Juniper Networks device maintains records of the most recent 50 probes for each configured test. These 50 probes are used to generate RPM statistics for a particular test.
Earliest Sample	System time when the first probe in the sample was received.	–
Latest Sample	System time when the last probe in the sample was received.	–
Mean Value	Average round-trip time for the 50-probe sample.	–

Table 127: Summary of Key RPM Output Fields *(Continued)*

Field	Values	Additional Information
Standard Deviation	Standard deviation of the round-trip times for the 50-probe sample.	–
Lowest Value	Shortest round-trip time from the device to the remote server, as measured over the 50-probe sample.	–
Time of Lowest Sample	System time when the lowest value in the 50-probe sample was received.	–
Highest Value	Longest round-trip time from the Juniper Networks device to the remote server, as measured over the 50-probe sample.	–
Time of Highest Sample	System time when the highest value in the 50-probe sample was received.	–
Cumulative Jitter for a Probe		
Samples	Total number of probes used for the data set.	The Juniper Networks device maintains records of the most recent 50 probes for each configured test. These 50 probes are used to generate RPM statistics for a particular test.
Earliest Sample	System time when the first probe in the sample was received.	–
Latest Sample	System time when the last probe in the sample was received.	–
Mean Value	Average jitter for the 50-probe sample.	–

Table 127: Summary of Key RPM Output Fields *(Continued)*

Field	Values	Additional Information
Standard Deviation	Standard deviation of the jitter values for the 50-probe sample.	–
Lowest Value	Smallest jitter value, as measured over the 50-probe sample.	–
Time of Lowest Sample	System time when the lowest value in the 50-probe sample was received.	–
Highest Value	Highest jitter value, as measured over the 50-probe sample.	–
Time of Highest Sample	System time when the highest jitter value in the 50-probe sample was received.	–

Example: Configuring Basic RPM Probes (SRX)

IN THIS SECTION

This example shows how to configure basic RPM probes to measure performance between two network endpoints.

Requirements

Before you begin:

- Establish basic connectivity.
- Configure network interfaces. See Interfaces User Guide for Security Devices.

Overview

In this example, you configure basic probes for two RPM owners, customerA and customerB. You configure the RPM test as icmp-test for customerA with a test interval of 15 seconds and specify a probe type as icmp-ping-timestamp, a probe timestamp, and a target address as 192.178.16.5. You then configure the RPM thresholds and corresponding SNMP traps to catch ingress (inbound) times greater than 3000 microseconds.

Then you configure the RPM test as http-test for customerB with a test interval of 30 seconds and specify a probe type as http-get and a target URL as http://customerB.net. Finally, you configure RPM thresholds and corresponding SNMP traps as probe-failure and test-failure to catch three or more successive lost probes and total lost probes of 10.

> **NOTE**: On SRX300, SRX320, SRX340, SRX1500 devices and vSRX Virtual Firewall instances, when you configure basic RPM probes, the following combination of the configuration parameters is not supported:
> Source address and destination port and next-hop.
>
> Configuring RPM probe with these parameters prevents sending out RPM probes to a specified probe target. We recommend you to configure either the source address or destination port and next-hop to configure RPM probe.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

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

```
set services rpm probe customerA test icmp-test probe-interval 15
set services rpm probe customerA test icmp-test probe-type icmp-ping-timestamp
set services rpm probe customerA test icmp-test hardware-timestamp
set services rpm probe customerA test icmp-test target address 192.178.16.5
set services rpm probe customerA test icmp-test thresholds ingress-time 3000
set services rpm probe customerA test icmp-test traps ingress-time-exceeded
set services rpm probe customerB test http-test probe-interval 30
set services rpm probe customerB test http-test probe-type http-get
set services rpm probe customerB test http-test target url http://customerB.net
set services rpm probe customerB test http-test thresholds successive-loss 3
set services rpm probe customerB test http-test thresholds total-loss 10
set services rpm probe customerB test http-test traps probe-failure
set services rpm probe customerB test http-test traps test-failure
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *Junos OS CLI User Guide* .

To configure basic RPM probes:

1. Configure the RPM.

```
[edit]
user@host# edit services rpm
```

2. Configure the RPM owners.

```
[edit services rpm]
user@host# set probe customerA
user@host# set probe customerB
```

3. Configure the RPM test for customerA.

```
[edit services rpm]
user@host# edit probe customerA
user@host# set test icmp-test probe-interval 15
user@host# set test icmp-test probe-type icmp-ping-timestamp
```

4. Specify a probe timestamp and a target address.

```
[edit services rpm probe customerA]
user@host# set test icmp-test hardware-timestamp
user@host# set test icmp-test target address 192.178.16.5
```

5. Configure RPM thresholds and corresponding SNMP traps.

```
[edit services rpm probe customerA]
user@host# set test icmp-test thresholds ingress-time 3000
user@host# set test icmp-test traps ingress-time-exceeded
```

6. Configure the RPM test for customerB.

```
[edit]
user@host# edit services rpm probe customerB
user@host# set test http-test probe-interval 30
```

7. Specify a probe type and a target URL.

```
[edit services rpm probe customerB]
user@host# set test http-test probe-type http-get
user@host# set test http-test target url http://customerB.net
```

8. Configure RPM thresholds and corresponding SNMP traps.

```
[edit services rpm probe customerB]
user@host# set test http-test thresholds successive-loss 3
user@host# set test http-test thresholds total-loss 10
user@host# set test http-test traps probe-failure
user@host# set test http-test traps test-failure
```

Results

From configuration mode, confirm your configuration by entering the `run show services rpm` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# run show services rpm
probe customerA {
     test icmp-test {
        probe-type icmp-ping-timestamp;
        target address 192.178.16.5;
        probe-interval 15;
        thresholds {
        ingress-time 3000;
         }
            traps ingress-time-exceeded;
             hardware-timestamp;
            }
        }
        probe customerB {
            test http-test {
            probe-type http-get;
            target url http://customerB.net;
            probe-interval 30;
         thresholds {
        successive-loss 3;
            total-loss 10;
            }
                traps [ probe-failure test-failure ];
        }
    }
```

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

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying RPM Services

Purpose

Verify that the RPM configuration is within the expected values.

Action

From operational mode, enter the `show services rpm` command. The output shows the values that are configured for RPM on the device.

Verifying RPM Statistics

Purpose

Verify that the RPM probes are functioning and that the RPM statistics are within expected values.

Action

From operational mode, enter the `show services rpm probe-results` command.

```
user@host> show services rpm probe-results
```

```
    Owner: customerD, Test: icmp-test
    Probe type: icmp-ping-timestamp
```

```
Minimum Rtt: 312 usec, Maximum Rtt: 385 usec, Average Rtt: 331 usec,
Jitter Rtt: 73 usec, Stddev Rtt: 27 usec
Minimum egress time: 0 usec, Maximum egress time: 0 usec,
Average egress time: 0 usec, Jitter egress time: 0 usec,
Stddev egress time: 0 usec
Minimum ingress time: 0 usec, Maximum ingress time: 0 usec,
Average ingress time: 0 usec, Jitter ingress time: 0 usec,
Stddev ingress time: 0 usec
Probes sent: 5, Probes received: 5, Loss percentage: 0

Owner: customerE, Test: http-test
Target address: 192.176.17.4, Target URL: http://customerB.net,
Probe type: http-get
Minimum Rtt: 1093 usec, Maximum Rtt: 1372 usec, Average Rtt: 1231 usec,
Jitter Rtt: 279 usec, Stddev Rtt: 114 usec
Probes sent: 3, Probes received: 3, Loss percentage: 0

 Owner: Rpm-Bgp-Owner, Test: Rpm-Bgp-Test-1
Target address: 10.209.152.37, Probe type: icmp-ping, Test size: 5 probes
Routing Instance Name: LR1/RI1
Probe results:
  Response received, Fri Oct 28 05:20:23 2005
  Rtt: 662 usec
Results over current test:
  Probes sent: 5, Probes received: 5, Loss percentage: 0
  Measurement: Round trip time
    Minimum: 529 usec, Maximum: 662 usec, Average: 585 usec,
    Jitter: 133 usec, Stddev: 53 usec
Results over all tests:
  Probes sent: 5, Probes received: 5, Loss percentage: 0
  Measurement: Round trip time
    Minimum: 529 usec, Maximum: 662 usec, Average: 585 usec,
    Jitter: 133 usec, Stddev: 53 usec
```

Configure the traps you want using the `set services rpm probe p1 test t1 traps` command.

If a trap is triggered, you can view the same in the log file named **messages** using the `show snmp log messages | match rmopd` command.

Possible Option	Set of values
egress-jitter-exceeded	Exceeded jitter in egress time threshold
egress-std-dev-exceeded	Exceeded egress time standard deviation threshold
egress-time-exceeded	Exceeded maximum egress time threshold
ingress-jitter-exceeded	Exceeded jitter in ingress time threshold
ingress-std-dev-exceeded	Exceeded ingress time standard deviation threshold
probe-failure	Successive probe loss threshold reached
rtt-exceeded	Exceeded maximum round trip time threshold
std-dev-exceeded	Exceeded round trip time standard deviation threshold
test-completion	Test completed
test-failure	Total probe loss threshold reached

Example: Configuring RPM Using TCP and UDP Probes (SRX Series Firewalls)

IN THIS SECTION

This example shows how to configure RPM using TCP and UDP probes.

Requirements

Before you begin:

- Establish basic connectivity.
- Configure network interfaces. See Interfaces User Guide for Security Devices.
- Configure the probe owner, the test, and the specific parameters of the RPM probe. See "Real-Time Performance Monitoring for SRX Devices" on page 687.

Overview

In this example, you configure both the host (device A) and the remote device (device B) to act as TCP and UDP servers. You configure a probe for customerC, which uses TCP packets. Device B is configured as an RPM server for both TCP and UDP packets, using an lt services interface as the destination interface, and ports 50000 and 50037, respectively.

> **CAUTION**: Use probe classification with caution, because improper configuration can cause packets to be dropped.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

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

```
{device A}
set services rpm probe customerC test tcp-test probe-interval 5
set services rpm probe customerC test tcp-test probe-type tcp-ping
```

```
set services rpm probe customerC test tcp-test target address 192.162.45.6
set services rpm probe customerC test tcp-test destination-interface lt-0/0/0
set services rpm probe customerC test tcp-test destination-port 50000
```

```
{device B}
set services rpm probe-server tcp port 50000
set services rpm probe-server udp port 50037
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *Junos OS CLI User Guide* .

To configure RPM using TCP and UDP probes:

1. Configure the RPM owner on device A.

```
{device A}
[edit]
user@host# edit services rpm
user@host# set probe customerC
```

2. Configure the RPM test.

```
{device A}
[edit services rpm]
user@host# edit services rpm probe customerC
user@host# set test tcp-test probe-interval 5
```

3. Set the probe type.

```
{device A}
[edit services rpm probe customerC]
user@host# set test tcp-test probe-type tcp-ping
```

4. Specify the target address.

```
{device A}
[edit services rpm probe customerC]
user@host# set test tcp-test target address 192.162.45.6
```

5. Configure the destination interface.

```
{device A}
[edit services rpm probe customerC]
user@host# set test tcp-test destination-interface It-0/0/0
```

6. Configure port 50000 as the TCP port to which the RPM probes are sent.

```
{device A}
[edit services rpm probe customerC]
user@host# set test tcp-test destination-port 50000
```

7. Configure device B to act as a TCP server using port 50000.

```
{device B}
[edit]
user@host# edit services rpm
user@host# set probe-server tcp port 50000
```

8. Configure device B to act as a UDP server using port 50037.

```
{device B}
[edit services rpm]
user@host# set probe-server udp port 50037
```

Results

From operational mode, confirm your configuration by entering the `show services rpm` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show services rpm
    probe customerC {
    test tcp-test {
        probe-type tcp-ping;
        target address 192.162.45.6;
        probe-interval 5;
        destination-port 50000;
        destination-interface lt-0/0/0.0;
    }
}
probe-server {
    tcp {
        port 50000;
    }
    udp {
        port 50037;
    }
}
```

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

Verification

IN THIS SECTION

Verifying RPM Probe Servers

Purpose

Confirm that the configuration is working properly.

Verify that the device is configured to receive and transmit TCP and UDP RPM probes on the correct ports.

Action

From operational mode, enter the `show services rpm active-servers` command. The output shows a list of the protocols and corresponding ports for which the device is configured as an RPM server.

```
user@host> show services rpm active-servers
```

```
    Protocol: TCP, Port: 50000

    Protocol: UDP, Port: 50037
```

Example: Configuring RPM Probes for BGP Monitoring

IN THIS SECTION

This example shows how to configure RPM probes to monitor BGP neighbors.

Requirements

Before you begin:

- Configure the BGP parameters under RPM configuration to send RPM probes to BGP neighbors. See "Real-Time Performance Monitoring for SRX Devices" on page 687.

- Use TCP or UDP probes by configure both the probe server (Juniper Networks device) and the probe receiver (the remote device) to transmit and receive RPM probes on the same TCP or UDP port. See "Real-Time Performance Monitoring for SRX Devices" on page 687.

Overview

In this example, you specify a hexadecimal value that you want to use for the data portion of the RPM probe as ABCD123. (It ranges from 1 through 2048 characters.) You specify the data size of the RPM probe as 1024 bytes. (The value ranges from 0 through 65,507.)

Then you configure destination port 50000 as the TCP port to which the RPM probes are sent. You specify the number of probe results to be saved in the probe history as 25. (It ranges from 0 through 255, and the default is 50.) You set the probe count to 5 and probe interval as 1. (The probe count ranges from 1 through 15, and the default is 1; and the probe interval ranges from 1 through 255, and the default is 3.) You then specify tcp-ping as the type of probe to be sent as part of the test.

Finally, you set the test interval as 60. The value ranges from 0 through 86,400 seconds for the interval between tests.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

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

```
set services rpm bgp data-fill ABCD123 data-size 1024
set services rpm bgp destination-port 50000 history-size 25
set services rpm bgp probe-count 5 probe-interval 1
set services rpm bgp probe-type tcp-ping test-interval 60
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *Junos OS CLI User Guide* .

To configure RPM probes to monitor BGP neighbors:

1. Configure the RPM and BGP.

```
[edit]
user@host# edit services rpm bgp
```

2. Specify a hexadecimal value.

```
[edit services rpm bgp]
user@host# set data-fill ABCD123
```

3. Specify the data size of the RPM probe.

```
[edit services rpm bgp]
user@host# set data-size 1024
```

4. Configure the destination port.

```
[edit services rpm bgp]
user@host# set destination-port 50000
```

5. Specify the number of probes.

```
[edit services rpm bgp]
user@host# set history-size 25
```

6. Set the probe count and probe interval.

```
[edit services rpm bgp]
user@host# set probe-count 5 probe-interval 1
```

7. Specify the type of probe.

```
[edit services rpm bgp]
user@host# set probe-type tcp-ping
```

NOTE: If you do not specify the probe type the default ICMP probes are sent.

8. Set the test interval.

```
[edit services rpm bgp]
user@host# set test-interval 60
```

Results

From configuration mode, confirm your configuration by entering the `run show services rpm` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
    user@host# run show services rpm
    bgp {
    probe-type tcp-ping;
        probe-count 5;
        probe-interval 1;
        test-interval 60;
        destination-port 50000;
        history-size 25;
        data-size 1024;
    data-fill ABCD123;
}
```

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

Verification

IN THIS SECTION

Verifying RPM Probes for BGP Monitoring

Purpose

Confirm that the configuration is working properly.

Verify that the RPM probes for BGP monitoring is configured.

Action

From operational mode, enter the `show services rpm` command.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
18.4R1	Starting in Junos OS Release 18.4R1, if the result of a probe or test exceeds the packet loss threshold, the real-time performance monitoring (RPM) test probe is marked as failed. The test probe also fails when the round-trip time (RTT) exceeds the configured threshold value. As a result, the device generates an SNMP notification (trap) and marks the RPM test as failed. RPM 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.

Configuring RPM Receiver Servers

The RPM TCP and UDP probes are proprietary to Juniper Networks and require a receiver to receive the probes. To configure a server to receive the probes, include the `probe-server` statement at the `[edit` `services rpm]` hierarchy level for Junos OS or the `[edit services monitoring rpm]` hierarchy level for Junos OS Evolved:

```
[edit services rpm]
probe-server {
    tcp {
        destination-interface interface-name;
        port number;
    }
    udp {
        port number;
    }
}
```

The port number specified for the UDP and TCP server can be `7` or from `49160` through `65535`.

> **NOTE**: The `destination-interface` statement is not supported on PTX Series Packet Transport routers, or for devices running Junos OS Evolved.

When you configure either `probe-type udp-ping` or `probe-type udp-ping-timestamp` along with the `one-way-hardware-timestamp` command, the value for the `destination-port` can be only 7. A constraint check prevents you for configuring any other value for the destination port in this case.

RELATED DOCUMENTATION

Understanding Using Probes for Real-Time Performance Monitoring on ACX, MX, and PTX Series Routers and EX and QFX Switches | **667**

Examples: Configuring Real-Time Performance Monitoring on MX, M, T and PTX Series Routers | **738**

Limiting the Number of Concurrent RPM Probes on M, MX, T and PTX Routers and EX Series Switches

To configure the maximum number of concurrent probes allowed, include the `probe-limit` statement at the `[edit services rpm]` hierarchy level:

```
probe-limit limit;
```

Specify a limit from `1` through `2000`. The default maximum number is 100.

Starting in Junos OS Release 17.2R2 and 17.3R1 for MX Series routers only, the probe-limit is 1 through 2000.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.2R2	Starting in Junos OS Release 17.2R2 and 17.3R1 for MX Series routers only, the probe-limit is 1 through 2000.

RELATED DOCUMENTATION

Understanding Using Probes for Real-Time Performance Monitoring on ACX, MX, and PTX Series Routers and EX and QFX Switches | 667

Examples: Configuring Real-Time Performance Monitoring on MX, M, T and PTX Series Routers | 738

Configuring RPM Timestamping on MX, M, T, and PTX Series Routers and EX Series Switches

To account for latency in the communication of probe messages, you can enable timestamping of the probe packets. You can timestamp the following RPM probe types: `icmp-ping`, `icmp-ping-timestamp`, `udp-ping`, and `udp-ping-timestamp`.

On M Series and T Series routers with an MS-PIC, on MX Series routers with an MS-DPC, MS-MIC, or MS-MPC linecard, on MX10000 Series routers, on PTX10008 and PTX10016 routers, and on EX Series switches, you can enable hardware timestamping of RPM probe messages. The timestamp is applied on both the RPM client device (the router or switch that originates the RPM probes) and the RPM probe server and applies only to IPv4 traffic. It is supported on the following:

- Layer 2 service package on MS-PICs, MS-DPCs, MS-MPCs, and MS-MICs.
- Layer 3 service package on MS-PICs, MS-DPCs, MS-MPCs, and MS-MICs.
- Extension-provider services package on M Series, MX Series, and T Series services PICs that support the Extension-Provider packages (In Junos OS releases earlier than Release 12.3, the extension-provider packages were variously referred to as Junos Services Framework (JSF), MP-SDK, and eJunos.)
- Layer 2, Layer 3, SDK Services, and PFE RPM timestamping interoperate with each other. Here, the RPM client can be on the Layer 3 `sp-` interface and the RPM server can be on an SDK Services package.

Two-way timestamping is available on `sp-` and `ms-` interfaces. To configure two-way timestamping on M Series and T Series routers, include the `destination-interface` statement at the `[edit services rpm probe` *`probe-owner`* `test` *`test-name`*`]` hierarchy level:

```
destination-interface sp-fpc/pic/port.logical-unit
destination-interface ms-fpc/pic/port.logical-unit
```

Specify the RPM client router and the RPM server router on the services logical interface or the multiservices interface by including the `rpm` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level:

```
rpm  (client | server);
```

The logical interface must be dedicated to the RPM task. It requires configuration of the `family inet` statement and a `/32` address, as shown in the example. This configuration is also needed for other services such as NAT and stateful firewall. You cannot configure RPM service on `unit 0` because RPM requires a dedicated logical interface; the same unit cannot support both RPM and other services. Because active flow monitoring requires `unit 0`, but RPM can function on any logical interface, a constraint check prevents you from committing an RPM configuration there.

On MX Series routers, on M320 Series routers using the Enhanced Queuing MPC, and on EX Series switches, you include the `hardware-timestamp` statement at the `[edit services rpm probe` *`probe-name`* `test` *`test-name`*`]` hierarchy level to specify that the probes are to be timestamped in the Packet Forwarding Engine host processor:

On MX Series routers, on MX10000 Series routers, on PTX5000, PTX10008, and PTX10016 routers, and on EX Series switches, you can include the `hardware-timestamp` statement at the `[edit services rpm probe`

probe-name test *test-name*] hierarchy level to specify that the probes are to be timestamped in the Packet Forwarding Engine host processor. On MX Series routers, hardware timestamping is supported on the following line cards:

- DPC
- DPCE
- MPC1
- MPC2
- MPC3
- MPC4
- MPC5
- MPC6
- MPC7

```
hardware-timestamp;
```

On the client side, these probes are timestamped in the Packet Forwarding Engine host processor on the egress DPC on the MX Series or M320 Series router or EX Series switch originating the RPM probes (RPM client). On the responder side (RPM server), the RPM probes to be timestamped are handled by the Packet Forwarding Engine host processor, which generates the response instead of the RPM process. The RPM probes are timestamped only on the router that originates them (RPM client). As a result, only round-trip time is measured for these probes.

When using the `hardware-timestamp` statement, the `data-size` value for the probe must be at least 100 bytes smaller than the default MTU of the interface of the RPM client interface (see "Configuring RPM Probes on M, MX and T Series Routers and EX Series Switches" on page 677). If hardware timestamping of RPM probe messages is enabled, the maximum data size that you can configure by using the data-size statement is limited to 1400.

NOTE: The Packet Forwarding Engine-based RPM feature does not support any stateful firewall configurations. If you need to combine RPM timestamping with a stateful firewall, use the interface-based RPM timestamping service described earlier in this section. MS-DPCs support stateful firewall processing as well as RPM timestamping.

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:

```
one-way-hardware-timestamp;
```

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.

For more information about these configurations, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

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

```
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 {
        term t1 {
            from {
                dscp af11;
            }
            then {
                forwarding-class assured-forwarding;
            }
        }
    }
}
interfaces sp-1/2/0 {
    unit 2 {
        rpm client;
        family inet {
            address 10.8.4.2/32;
            filter {
                input f1;
            }
        }
    }
}
interfaces sp-1/2/1 {
    unit 2 {
        rpm server;
        family inet {
            address 10.8.3.2/32;
            filter {
                input f1;
            }
        }
    }
}
```

For more information about firewall filters, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide; for more information about queuing, see the Class of Service User Guide (Routers and EX9200 Switches).

RELATED DOCUMENTATION

Understanding Using Probes for Real-Time Performance Monitoring on ACX, MX, and PTX Series Routers and EX and QFX Switches | 667

Examples: Configuring Real-Time Performance Monitoring on MX, M, T and PTX Series Routers | 738

Configuring the Interface for RPM Timestamping for Client/Server on a Switch (EX Series)

Use real-time performance monitoring (RPM) to configure active probes to track and monitor traffic across the network and to investigate network problems. To configure basic RPM probes on the EX Series or QFX Series switch, you must configure the probe owner, the test, and the specific parameters of the RPM probe.

You can also set a timestamp to improve the measurement of latency or jitter. The probe is timestamped by the device originating the probe (the RPM client). If you do not enable hardware timestamps, the timer values are set. You should configure both the RPM client (the requester) and the RPM server (the responder) to timestamp the RPM packets. However, if the RPM server does not support hardware timestamps, RPM can only report the round-trip measurements.

> **NOTE**: On the EX4300 switch, RPM timestamping is performed in the software. The RPM probes at the requester and responder devices are timestamped in the Packet Forwarding Engine instead of the Junos OS process (rmpod) that runs on the Routing Engine. This timestamping method is referred to as pseudo-hardware timestamping.

> **NOTE**: QFX Series switches do not support hardware timestamps.

Timestamps apply only to IPv4 traffic.

You can enable hardware timestamps for the following RPM probe types:

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

To configure RPM probes and to enable hardware timestamping:

1. Specify the probe owner:

```
[edit services rpm]
user@switch# set probe owner
```

2. Specify a test name. A test represents the range of probes over which the standard deviation, average, and jitter are calculated.

```
[edit services rpm probe owner]
user@switch# set test test-name
```

3. Specify the packet and protocol contents of the probe:

```
[edit services rpm probe owner test test-name]
user@switch# set probe-type type
```

4. Specify the destination IPv4 address to be used for the probes:

```
[edit services rpm probe owner test test-name]
user@switch# set target address
```

5. Specify the number of probes within a test:

```
[edit services rpm probe owner test test-name]
user@switch# set probe-count count
```

6. Specify the time, in seconds, to wait between sending packets:

```
[edit services rpm probe owner test test-name]
user@switch# set probe-interval interval
```

7. Specify the time, in seconds, to wait between tests:

```
[edit services rpm probe owner test test-name]
user@switch# set test-interval interval
```

8. Specify the source IP address to be used for probes. If the source IP address is not one of the switch's assigned addresses, the packet uses the outgoing interface's address as its source.

```
[edit services rpm probe owner test test-name]
user@switch# set source-address address
```

9. Specify the value of the Differentiated Services (DiffServ) field within the IP header. The DiffServ code point (DSCP) bits value must be set to a valid 6-bit pattern.

```
[edit services rpm probe owner test test-name]
user@switch# set dscp-code-points dscp-bits
```

10. If you are using ICMP probes, specify the size of the data portion of ICMP probes:

```
[edit services rpm probe owner test test-name]
user@switch# set data-size size
```

11. Enable hardware timestamping of RPM probe messages:

NOTE: QFX Series switches do not support hardware timestamps.

```
[edit services rpm probe owner test test-name]
user@switch# set hardware-timestamp
```

Analyzing Network Efficiency in IPv6 Networks on MX Series Routers Using RPM Probes

IN THIS SECTION

Real-time performance monitoring (RPM) is a mechanism that 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 OS that enables a router to measure metrics such as round-trip delays and unanswered echo requests. To compute these parameters, 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 such as 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 obtained from the results of the RPM test. 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.

Starting with Junos OS release 16.1, the RPM client router (the router or switch that originates the RPM probes) can send probe packets to the RPM probe server (the device that receives the RPM probes) that contains an IPv6 address. To specify the destination IPv6 address used for the probes, include the `target (url` *`ipv6-url`* `| address` *`ipv6-address`*`)` statement at the `[edit services rpm probe owner test` *`test-name`*`]` hierarchy level. The protocol family for IPv6 is named inet6.

```
[edit services rpm]
probe owner {
    test test-name {
        target (url ipv6-url | address ipv6-address);
    }
}
```

To specify the IPv6 protocol-related settings and the source IPv6 address of the client from which the RPM probes are sent, include the `inet6-options source-address` *`ipv6-address`* statement at the `[edit services rpm probe owner test` *`test-name`*`]` hierarchy level. A probe request is a standard packet with corresponding TCP, UDP, and ICMP headers over the IPv6 header. No RPM header is appended to the standard packet for Routing Engine-based RPM implementation. A probe response is also a standard packet with corresponding TCP, UDP, and ICMP headers over the IPv6 header. No RPM header is appended to the standard packet for Routing Engine-based RPM implementation.

```
[edit services rpm]
probe owner {
    test test-name {
        inet6-options source-address ipv6-address;
    }
}
```

The output of the `show services rpm probe-results owner` *`probe-name`* `test` *`test-name`* and `show services rpm history-results owner` *`owner`* `test` *`name`* commands that display the results of the most recent RPM probes and results of historical RPM probes respectively have been enhanced to display the target address as IPv6 address and other IPv6 information for probes sent to IPv6 servers or destinations. The existing SNMP Get requests and traps for IPv6 are applicable for IPv6 probes. The target type field in the SNMP set operation contains IPv6 source and destination addresses.

Guidelines for Configuring RPM Probes for IPv6 Destinations

Keep the following points in mind when you configure IPv6 addresses for RPM destinations or servers:

- Only Routing Engine-based RPM is supported for IPv6 targets including VRF support, specification of the size of the data portion of ICMP probes, data pattern, and traffic class.
- You can configure probes with a combination of IPv4 and IPv6 tests. However, a test can be either IPv4 or IPv6-based at a point in time. The OS impacts the accuracy of the measurements because the variability factor introduced by the general OS that performs the system processing proved is significantly larger than the amount of time spent by the packet traversing on the wire. This condition causes round-trip time (RTT) spikes to be seen even with a single test.
- Routing Engine-based RPM does not support one-way hardware-based timestamping.
- One-way measurements are not supported here because timestamping is done only on the RPM client side.
- The maximum number of concurrent probes allowed (by including the `probe-limit` statement at the `[edit services rpm]` hierarchy level) is 1000. We recommend that the limit on concurrent probes be set as 10. Higher concurrent probes can result in higher spikes. The maximum number of tests you can configure is 1000. RPM cannot be configured on logical systems. SNMP set operation is permitted only on ICMP probes and it is not supported for other type of probes.
- The `hardware-timestamp` and `one-way-hardware-timestamp` statements at the `[edit services rpm probe owner test` *`test-name`*`]` hierarchy level are not supported for IPv6.
- You cannot specify the `icmp-ping` (which sends ICMP echo requests to a target address) and the `icmp-ping-timestamp` (which sends ICMP timestamp requests to a target address) options with the probe-type statement at the `[edit services rpm probe owner test` *`test-name`*`]` hierarchy level.
- Some of the RPM problems can resolved by restarting the SNMP remote operations process (rmopd) on the Routing Engine by using the restart remote-operations command. If RPM needs to be disabled, the rpm statement at the [edit services] hierarchy level needs to be deleted or deactivated. PIC, Packet Forwarding Engine, and lookup chip (LU) based RPM implementation for IPv6 are not supported.
- The following table describes the IPv6 special address prefixes that are not supported.

IPV6 Address Type	IPV6 Address Prefix
Node-Scoped Unicast	::1/128 is the loopback address ::/128 is the unspecified address
IPv4-Mapped Addresses	::FFFF:0:0/96
IPv4-Compatible Addresses	:<ipv4-address>/96
Link-Scoped Unicast	fe80::/10
Unique-Local	fc00::/7
Documentation Prefix	2001:db8::/32
6to4	2002::/16
6bone	5f00::/8
ORCHID	2001:10::/28
Teredo	2001::/32
Default Route	::/0
Multicast	ff00::/8

- The current scaling number for IPv4 probes is a maximum of 500 concurrent probes and the limit on the maximum number of configurable tests is 1000. These scaling parameters are applicable for IPv6 probes. The same scaling limits are applicable, even in cases where both IPv4-based tests and IPv6-based tests are run at the same time.

- The minimum rate of probes is 1 probe per second and the maximum interval between tests is 86400 seconds. These scaling and performance numbers vary based on whether the Two-Way Active Measurement Protocol (TWAMP) server and client are configured on the same router. This condition occurs because the TWAMP server/client has packet processing in RMOPD and it competes with

RPM functionality in the same process. The RTT of IPv6-based RPM and ping utilities must be equivalent for data size. In Routing Engine-based RPM implementation, RTT spikes are seen owing to various queuing delays introduced in the system. This behavior can be noticed even with a single test.

- Some of the TCP and UDP ports might be opened to communicate between the RPM server and RPM client. Therefore, we recommend that you use firewalls and distributed denial-of-service (DDoS) attack filters to ensure that no security threats are possible by some third-party attackers or hackers.

- The different packet types that can be used within the probe include:

  - ICMP6 echo
  - UDP echo
  - UDP timestamp

RELATED DOCUMENTATION

Configuring RPM Probes on MX Series Routers and EX Series Switches | 677

Configuring BGP Neighbor Discovery Through RPM

BGP neighbors can be configured at the following hierarchy levels:

- `[edit protocols bgp group` *group-name*`]`—Default logical system and default routing instance.

- `[edit routing-instances` *instance-name* `protocols bgp group` *group-name*`]`—Default logical system with a specified routing instance.

- `[edit logical-systems` *logical-system-name* `protocols bgp group` *group-name*`]`—Configured logical system and default routing instance.

- `[edit logical-systems` *logical-system-name* `routing-instances` *instance-name* `protocols bgp group` *group-name*`]`—Configured logical system with a specified routing instance.

If a device has a large number of BGP neighbors configured, you can direct (filter) the RPM probes to a selected group of BGP neighbors rather than to all the neighbors. To identify the BGP devices to receive RPM probes, you can configure routing instances.

When you configure BGP neighbor discovery through RPM, if you do not specify a logical system, the RPM probe applies to configured BGP neighbors for all logical systems. If you do not specify a routing instance, the RPM probe applies to configured BGP neighbors in all routing instances. You can explicitly configure RPM probes to apply only to the default logical system, the default routing instance, or to a particular logical system or routing instance.

To configure BGP neighbor discovery through RPM, configure the probe properties at the `[edit services rpm bgp]` hierarchy:

```
data-fill data;
data-size size;
destination-port port;
history-size size;
logical-system logical-system-name [routing-instances routing-instance-name];
moving-average-size number;
probe-count count;
probe-interval seconds;
probe-type type;
routing-instances instance-name;
test-interval interval;
```

- To specify the contents of the data portion of Internet Control Message Protocol (ICMP) probes, include the `data-fill` statement at the `[edit services rpm bgp]` hierarchy level. The value can be a hexadecimal value.
- To specify the size of the data portion of ICMP probes, include the `data-size` statement at the `[edit services rpm bgp]` hierarchy level. The size can be from `0` through `65400` and the default size is `0`.
- To specify the User Datagram Protocol (UDP) port or Transmission Control Protocol (TCP) port to which the probe is sent, include the `destination-port` statement at the `[edit services rpm bgp]` hierarchy level. The `destination-port` statement is used only for the UDP and TCP probe types. The value can be `7` or from `49160` through `65535`.
- To specify the number of stored history entries, include the `history-size` statement at the `[edit services rpm bgp]` hierarchy level. Specify a value from `0` to `512`. The default is `50`.
- To specify the logical system used by ICMP probes, include the `logical-system` *`logical-system-name`* statement at the `[edit services rpm bgp]` hierarchy level. If you do not specify a logical system, the RPM probe applies to configured BGP neighbors for all logical systems. To apply the probe to only the default logical system, you must set the value of *`logical-system-name`* to `null`.
- To specify a number of samples for making statistical calculations, include the moving-average-size statement at the `[edit services rpm bgp]` hierarchy level. Specify a value from 0 through 255.
- To specify the number of probes within a test, include the `probe-count` statement at the `[edit services rpm bgp]` hierarchy level. Specify a value from `1` through `15`.

- To specify the time to wait between sending packets, include the `probe-interval` statement at the `[edit services rpm bgp]` hierarchy level. Specify a value from `1` through `255` seconds.
- To specify the packet and protocol contents of the probe, include the `probe-type` statement at the `[edit services rpm bgp]` hierarchy level. The following probe types are supported:
  - `icmp-ping`—Sends ICMP echo requests to a target address.
  - `icmp-ping-timestamp`—Sends ICMP timestamp requests to a target address.
  - `tcp-ping`—Sends TCP packets to a target.
  - `udp-ping`—Sends UDP packets to a target.
  - `udp-ping-timestamp`—Sends UDP timestamp requests to a target address.

  > **NOTE**: Some probe types require additional parameters to be configured. For example, when you specify the `tcp-ping` or `udp-ping` option, you must configure the destination port using the `destination-port` *`port`* statement. The `udp-ping-timestamp` option requires a minimum data size of 12; any smaller data size results in a commit error. The minimum data size for TCP probe packets is 1.

- To specify the routing instance used by ICMP probes, include the `routing-instances` statement at the `[edit services rpm bgp]` hierarchy level. The default routing instance is Internet routing table `inet.0`. If you do not specify a routing instance, the RPM probe applies to configured BGP neighbors in all routing instances. To apply the RPM probe to only the default routing instance, you must explicitly set the value of *`instance-name`* to `default`.
- To specify the time to wait between tests, include the `test-interval` statement at the `[edit services bgp probe]` hierarchy level. Specify a value from `0` through `86400` seconds. A value of 0 seconds causes the RPM test to stop after one iteration. The default value is 1.

RELATED DOCUMENTATION

Understanding Using Probes for Real-Time Performance Monitoring on ACX, MX, and PTX Series Routers and EX and QFX Switches | **667**

Examples: Configuring BGP Neighbor Discovery on SRX Series Firewalls and MX, M, T and PTX Series Routers With RPM | **732**

Examples: Configuring BGP Neighbor Discovery on SRX Series Firewalls and MX, M, T and PTX Series Routers With RPM

If a device has a large number of BGP neighbors configured, you can direct (filter) the RPM probes to a selected group of BGP neighbors rather than to all the neighbors. To identify the BGP devices to receive RPM probes, you can configure routing instances.

Configure BGP neighbor discovery with RPM for all logical systems and all routing instances:

```
[edit services rpm]
bgp {
    probe-type icmp-ping;
    probe-count 5;
    probe-interval 1;
    test-interval 60;
    history-size 10;
    data-size 255;
    data-fill 0123456789;
}
```

Configure BGP neighbor discovery with RPM for only the following logical systems and routing instances: `LS1/RI1`, `LS1/RI2`, `LS2`, and `RI3`:

```
[edit services rpm]
bgp {
    probe-type icmp-ping;
    probe-count 5;
    probe-interval 1;
    test-interval 60;
    history-size 10;
    data-size 255;
    data-fill 0123456789;
    logical-system {
        LS1 {
            routing-instances {
                RI1;
                RI2;
            }
        }
        LS2;
    }
```

```
    routing-instance {
        RI3;
    }
}
```

> **NOTE**: The `logical-system` statement is not supported on PTX Series Packet Transport routers.

Configure BGP neighbor discovery with RPM for only the default logical system and default routing instance:

```
[edit services rpm]
bgp {
    probe-type icmp-ping;
    probe-count 5;
    probe-interval 1;
    test-interval 60;
    history-size 10;
    data-size 255;
    data-fill 0123456789;
    logical-system {
        null {
            routing-instances {
                default;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Understanding Using Probes for Real-Time Performance Monitoring on ACX, MX, and PTX Series Routers and EX and QFX Switches | **667**

Configuring BGP Neighbor Discovery Through RPM | **729**

Trace RPM Operations

IN THIS SECTION

RPM tracing operations track all RPM operations and record them in a log file. The logged error descriptions provide detailed information to help you solve problems faster.

RPM Trace Operations Overview

In Junos OS, you enable tracing operations by configuring the traceoptions statement at the specific hierarchy level you want to trace. Junos OS Evolved uses a different tracing architecture. All running applications create trace information, with multiple instances of the same application having their own trace information. Therefore, in Junos OS Evolved, trace messages are logged, viewed, and configured by application. As a result, Junos OS Evolved does not support the `traceoptions` statement at many of the hierarchy levels that Junos OS supports.

In Junos OS Evolved, you do not view trace files directly, and you should never add, edit, or remove trace files under the **/var/log/traces** directory because this can corrupt the traces. Instead, you use the `show trace application` *`application-name`* `node` *`node-name`* command to read and decode trace messages stored in the trace files. All running applications on Junos OS Evolved create trace information at the `info` level by default.

In Junos OS, by default, no events are traced. You can change this default behavior by using the `traceoptions` statement. If you include the `traceoptions` statement at the `[edit services rpm]` hierarchy level, the default tracing behavior is the following:

- Important events are logged in a file called **rmopd** located in the **/var/log** directory.
- When the log file reaches 128 kilobytes (KB), it is renamed **rmopd.0**, then **rmopd.1**, and so on, until there are three trace files. Then the oldest trace file (**rmopd.2**) is overwritten.
- Log files can be accessed only by the user who configures the tracing operation.

RPM is governed by the `rmopd` application. For Junos OS Evolved, to configure traces for a severity other than `info` for the `rmopd` application, include the `application rmopd node` *`node-name`* `level` *`severity`* statement at the [edit system trace] hierarchy level.

> **NOTE**: For general monitoring and troubleshooting of devices running Junos OS or Junos OS Evolved, we recommend using standard tools such as CLI show commands, system log messages, SNMP, and telemetry data. You should avoid using trace messages for general debugging purposes and long-term solutions because they are subject to change without notice.

Configure the Trace Operations

By default, for Junos OS, if the `traceoptions` configuration is present, only important events are logged. You can configure the trace operations to be logged by including the following statements at the `[edit services rpm traceoptions]` hierarchy level:

```
flag {
    all;
    configuration;
    error;
    ipc;
    ppm;
    rpd;
    statistics
}
```

Table 128 on page 735 describes the meaning of the RPM tracing flags.

Table 128: Junos OS RPM Tracing Flags

Flag	Description	Default Setting
`all`	Trace all operations.	Off
`configuration`	Trace configuration events.	Off
`error`	Trace events related to catastrophic errors in daemon.	Off

Table 128: Junos OS RPM Tracing Flags *(Continued)*

Flag	Description	Default Setting
`ipc`	Trace IPC events.	Off
`ppm`	Trace ppm events.	Off
`rpd`	Trace rpd events.	Off
`statistics`	Trace statistics.	Off

By default, for Junos OS Evolved, all running applications create trace information at the `info` level. To configure traces for a severity other than `info` for the `rmopd` application, include the `application rmopd node` *`node-name`* `level` *`severity`* statement at the [edit system trace] hierarchy level. For information about the various configurable severity levels for Junos OS Evolved, see *trace (System)*.

SEE ALSO

Understanding Using Probes for Real-Time Performance Monitoring on ACX, MX, and PTX Series Routers and EX and QFX Switches | 667

Configure the RPM Log File Name

(Junos OS only) By default, the name of the file that records RPM trace output is **rmopd**. To specify a different file name:

```
[edit services rpm traceoptions]
user@host set file filename
```

Configure the Number and Size of RPM Log Files

(Junos OS only) To configure the limits on the number and size of RPM trace files:

```
[edit services rpm traceoptions]
user@host set file filename files number size size
```

The number of files can be from 2 through 1000 files. The file size of each file can be from 10 KB through 1 gigabyte (GB).

For example, set the maximum file size to 2 MB, and the maximum number of files to 20 for a log file named **rpmtrace**:

```
[edit services rpm traceoptions]
user@host set file rpmtrace files 20 size 2MB
```

When the **rpmtrace** file reaches 2 MB, it is renamed **rpmtrace.0**, and a new file called **rpmtrace** is created. When the new **rpmtrace** reaches 2 MB, **rpmtrace.0** is renamed **rpmtrace.1** and **rpmtrace** is renamed **rpmtrace.0**. This process repeats until there are 20 trace files. Then the oldest file (**rpmtrace.19**) is overwritten by **rpmtrace.18**.

Configure Access to the Log File

(Junos OS only) By default, log files can be accessed only by the user who configures the tracing operation.

To specify that any user can read all log files:

```
[edit services rpm traceoptions]
user@host set file filename world-readable
```

To explicitly set the default behavior:

```
[edit services rpm traceoptions]
user@host set file filename no-world-readable
```

Configure a Regular Expression for Lines to Be Logged

(Junos OS only) By default, the trace operation output includes all lines relevant to the logged events.

To refine the output by specifying a regular expression (regex) to be matched:

```
[edit services rpm traceoptions]
user@host set file filename match regular-expression
```

Examples: Configuring Real-Time Performance Monitoring on MX, M, T and PTX Series Routers

Configure an RPM instance identified by the probe name `probe1` and the test name `test1`:

```
[edit services rpm]
probe probe1{
    test test1 {
        dscp-code-points 001111;
        probe-interval 1;
        probe-type icmp-ping;
        target address 172.17.20.182;
        test-interval 20;
        thresholds rtt 10;
        traps rtt-exceeded;
    }
}
probe-server {
    tcp {
        destination-interface lt-0/0/0.0
        port 50000;
    }
    udp {
        destination-interface lt-0/0/0.0
        port 50001;
    }
}
probe-limit 200;
```

Configure packet classification, using `lt-` interfaces to send the probe packets to a logical tunnel input interface. By sending the packet to the logical tunnel interface, you can configure regular and multifield classifiers, firewall filters, and header rewriting for the probe packets. To use the existing tunnel framework, the `dlci` and `encapsulation` statements must be configured.

```
[edit 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;
        source-address 10.8.4.2;
        dscp-code-points ef;
        data-size 100;
        destination-interface lt-0/0/0.0;
    }
}
[edit interfaces]
lt-0/0/0 {
    unit 0 {
        encapsulation frame-relay;
        dlci 10;
        peer-unit 1;
        family inet;
    }
    unit 1 {
        encapsulation frame-relay;
        dlci 10;
        peer-unit 0;
        family inet;
    }
}
[edit class-of-service]
interfaces {
    lt-0/0/0 {
        unit 1 {
            classifiers {
                dscp default;
            }
        }
    }
}
```

Configure an input filter on the interface on which the RPM probes are received. This filter enables prioritization of the received RPM packets, separating them from the regular data packets received on the same interface.

```
[edit firewall]
filter recos {
    term recos {
        from {
```

```
            source-address {
                10.8.4.1/32;
            }
            destination-address {
                10.8.4.2/32;
            }
        }
        then {
            loss-priority high;
            forwarding-class network-control;
        }
    }
}
[edit interfaces]
fe-5/0/0 {
    unit 0 {
        family inet {
            filter {
                input recos;
            }
            address 10.8.4.2/24;
        }
    }
}
```

Configure an RPM instance and enable RPM for the extension-provider packages on the adaptive services interface:

```
[edit services rpm]
probe probe1{
    test test1 {
        data-size 1024;
        data-fill 0;
        destination-interface ms-1/2/0.10;
        dscp-code-points 001111;
        probe-count 10;
        probe-interval 1;
        probe-type icmp-ping;
        target address 172.17.20.182;
        test-interval 20;
        thresholds rtt 10;
        traps rtt-exceeded;
```

```
    }
}
[edit interfaces]
ms-1/2/0 {
    unit 0 {
        family inet;
    }
    unit 10 {
        rpm client;
        family inet {
            address 192.0.2.1/32;
        }
}
[edit chassis]
fpc 1 {
    pic 2 {
        adaptive-services {
            service-package {
                extension-provider {
                    control-cores 1;
                    data-cores 1;
                    object-cache-size 512;
                    policy-db-size 64;
                    package jservices-rpm;
                    syslog {

                        daemon any;
                    }
                }
            }
        }
    }
}
```

Configure the minimum statements necessary to enable TWAMP:

```
[edit services]
rpm {
    twamp {
        server {
            authentication-mode none;
            port 10000;                       # Twamp server's listening port
```

```
            client-list LIST-1 {              # LIST-1 is the name of the client-list. Multiple
lists can be configured.
                address {
                    198.51.100.2/30;              # IP address of the control client.
                }
            }
        }
    }
[edit interfaces sp-5/0/0]
unit 0 {
    family inet;
}
unit 10 {
    rpm {
        twamp-server;                       # You must configure a separate logical interface on
the service PIC interface for the TWAMP server.
    }
    family inet {
        address 203.0.113.50/32;             # This address must be a host address with a 32-bit
mask.
    }
}
[edit chassis]
fpc 5 {
    pic 0 {
        adaptive-services {
            service-package layer-2;      # Configure the service PIC to run in Layer 2 mode.
        }
    }
}
```

Configure additional TWAMP settings:

```
[edit services]
rpm {
    twamp {
        server {
            maximum-sessions 5;
            maximum-sessions-per-connection 2;
            maximum-connections 3;
            maximum-connections-per-client 1;
            port 10000;
```

```
            server-inactivity-timeout ;
            client-list LIST-1 {
                address {
                    198.51.100.2/30;
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

Understanding Using Probes for Real-Time Performance Monitoring on ACX, MX, and PTX Series Routers and EX and QFX Switches | **667**

Examples: Configuring BGP Neighbor Discovery on SRX Series Firewalls and MX, M, T and PTX Series Routers With RPM | **732**

Enabling RPM on MX, M and T Series Routers and SRX Firewalls for the Services SDK

Real-time performance monitoring (RPM), which has been supported on the adaptive services interface, is now supported by the Services SDK. RPM is supported on all platforms and service PICs that support the Services SDK.

To enable RPM for the Junos OS extension-provider package on the adaptive services interface, configure the `object-cache-size`, `policy-db-size`, and `package` statements at the `[edit chassis fpc *slot-number* pic *pic-number* adaptive-services service-package extension-provider]` hierarchy level. For the extension-provider package, *package-name* in the `package *package-name*` statement is `jservices-rpm`.

For more information about the extension-provider package, see the *SDK Applications Configuration Guide and Command Reference*.

The following example shows how to enable RPM for the extension-provider package on the adaptive services interface:

```
chassis fpc 1 {
    pic 2 {
        adaptive-services {
            service-package {
```

```
                extension-provider {
                    control-cores 1;
                    data-cores 1;
                    object-cache-size 512;
                    policy-db-size 64;
                    package jservices-rpm;
                    syslog daemon any;
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

Understanding Using Probes for Real-Time Performance Monitoring on ACX, MX, and PTX Series Routers and EX and QFX Switches | **667**

Examples: Configuring Real-Time Performance Monitoring on MX, M, T and PTX Series Routers | **738**

destination-interface

Understand Two-Way Active Measurement Protocol

SUMMARY

Learn about using Two-Way Active Measurement Protocol (TWAMP) to measure network performance between any two devices in a network.

IN THIS SECTION

The Two-Way Active Management Protocol (TWAMP), described in RFC 5357, is an extension of the One-Way Active Management Protocol (OWAMP) that supplies two-way or round-trip measurements instead of unidirectional capabilities. Two-way measurements are helpful because round-trip delays do not require host clock synchronization and remote support might be a simple echo function. However, the Internet Control Message Protocol (ICMP) Echo Request/Reply (used by ping) for this purpose has several shortcomings. TWAMP defines an open protocol for measuring two-way or round-trip metrics with greater accuracy than other methods by using time-stamps (processing delays can be factored as well).

Use Feature Explorer: Two-Way Active Measurement Protocol to confirm platform and release support for specific features.

Review the "Platform-Specific TWAMP Behavior" on page 750 section for notes related to your platform.

Benefits of TWAMP

- TWAMP probe configuration helps you activate, test, monitor, and troubleshoot your network end-to-end without using a dedicated testing device.
- TWAMP timestamps provide two-way or round-trip metrics with greater accuracy than other methods (processing delays can be factored as well).
- TWAMP is often used to check service-level agreement (SLA) compliance, and the TWAMP feature is often used in that context.
- Two-way measurements are better than one-way measurements because round-trip delays do not require host clock synchronization. This is possible because the reflector places its own sequence number in the packet.

> **NOTE**: We recommend that you do not configure the RPM client and a TWAMP server on the same device. This might cause some issues in the RPM probe results.

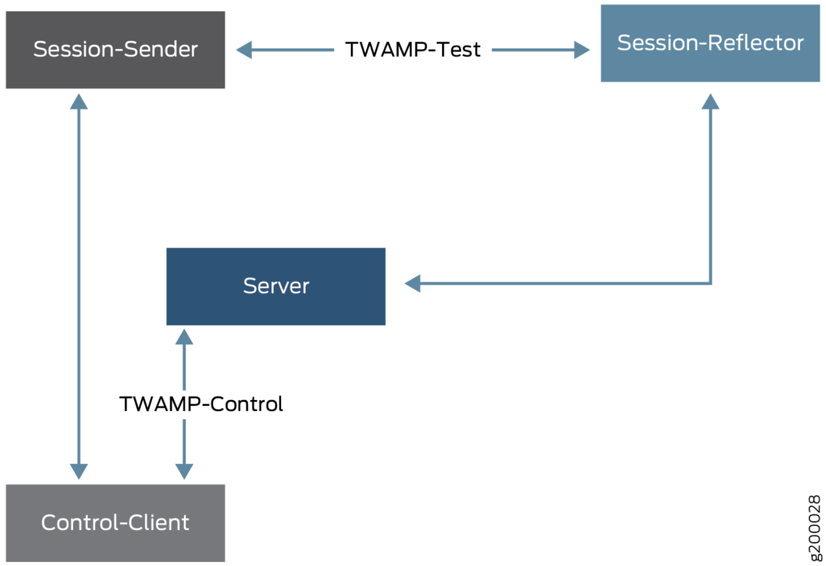
Understand Two-Way Active Measurement Protocol (TWAMP)

Usually, TWAMP operates between interfaces on two devices playing specific roles. TWAMP is often used to check Service Level Agreement (SLA) compliance, and the TWAMP feature is often presented in that context. TWAMP uses two related protocols, running between several defined elements:

- TWAMP-Control—Initiates, starts, and ends test sessions. The TWAMP-Control protocol runs between a Control-Client element and a Server element.
- TWAMP-Test—Exchanges test packets between two TWAMP elements. The TWAMP-Test protocol runs between a Session-Sender element and a Session-Reflector element.

The four elements are shown in Figure 64 on page 746:

Figure 64: Four Elements of TWAMP



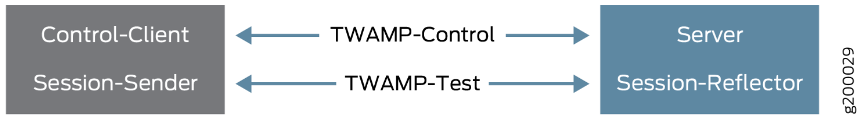
Although four different TWAMP devices can perform the four logical roles of TWAMP Control-Client, Server, Session-Sender, and Session-Reflector, different devices can play different roles. A common implementation combines the roles of Control-Client and Session-Sender in one device (known as the TWAMP controller or TWAMP client) and the roles of Server and Session-Reflector in the other device (known as the TWAMP responder or TWAMP server). In this case, each device runs both the TWAMP-Control (between Control-Client and Server) and TWAMP-Test (between Session-Sender and Session-Reflector) protocols.

The TWAMP client-server architecture as implemented looks like this:

- TWAMP client
  - Control-Client sets up, starts and stops the TWAMP test sessions.
  - Session-Sender creates TWAMP test packets that are sent to the Session-Reflector in the TWAMP server.
- TWAMP server
  - Session-Reflector sends back a measurement packet when a test packet is received, but does not maintain a record of such information.
  - Server manages one or more sessions with the TWAMP client and listens for control messages on a TCP port.

The packaging of these elements into TWAMP client and TWAMP server processes is shown in Figure 65 on page 747.

Figure 65: The Elements of TWAMP Implemented as Client (Left) and Server (Right).



TWAMP Light Support

TWAMP Light, as defined in Appendix I of RFC 5357, is a stateless version of TWAMP where test parameters are predefined instead of negotiated. All test packets received by the server on a test port are reflected back and forgotten right away. For those devices that support them, you can also configure IPv6 target addresses, including IPv6 link-local target addresses.

Use Feature Explorer: Two-Way Active Measurement Protocol to confirm platform and release support for specific features.

Simple Two-Way Active Measurement Protocol (STAMP) Support

As defined in RFC 8762, *Simple Two-Way Active Measurement Protocol*, STAMP standardizes and expands upon the TWAMP Light operational mode, which was defined in Appendix I of RFC 5357, *Two-Way Active Measurement Protocol* (TWAMP). For those devices that support STAMP, a STAMP-compliant reflector ensures symmetric payload size (in accordance with RFC 6038) and operates in either stateless or stateful mode, depending on whether the sequence number in the reflected payload is copied from the client frame or generated independently. A stateful reflector can detect in which direction drops have occurred. In previous releases, we supported symmetric payloads and stateless reflection. We support stateful reflection, full compliance with the STAMP standard, and unidirectional drop values for clients. We support unidirectional drop values not only for STAMP clients, but also for TWAMP-Managed-mode clients. For Junos OS Evolved, STAMP is configured at the `[edit services monitoring twamp server light]` hierarchy level. Stateful reflection is configured with the `stateful-sequence` statement. For servers, the new default for `offload-type` is now `pfe-timestamp` instead of `inline-timestamp`.

Use Feature Explorer: Simple Two-Way Active Measurement Protocol (STAMP) to confirm platform and release support.

Static Route Tracking

Starting in Junos OS Evolved Release 24.4R1, we've extended support for static route tracking to Junos OS Evolved and included Two-Way Active Measurement Protocol (TWAMP) test support as well. You use TWAMP probes to detect link status and to change the preferred-route state on the basis of the probe results. Tracked static routes can be IPv4 or IPv6, and each IPv4 and IPv6 tracked static route supports up to 16 next hops. You can also configure the metric, route preference, and tag values for each IPv4 or IPv6 destination prefix. However, you configure this feature differently on Junos OS Evolved devices; you configure the `sla-tracking` statement at the `[edit routing-options]` hierarchy level. You also use a different command, `show route sla-tracking`, to see information about these routes.

Segment Routing Support for Configuring Paths

Starting in Junos OS Evolved Release 24.4R1 for those PTX routers that support it and in Junos OS Release 25.4R1 for those MX routers that support it, we have added support in Two-Way Active Measurement Protocol (TWAMP) for segment routing (SR) as defined in RFC 8402, which broadly specifies the SR architecture. We support two types of SR for TWAMP probes:

- **SR-MPLS:** Uses a list of labels, each representing a segment end node.
- **SRv6:** Uses a type 4 IPv6 routing header introduced in RFC 8754, with each segment end node represented as an IPv6 address or IPv6 segment identifier (SID).

You can specify the list of SR-MPLS or SRv6 segments for a TWAMP probe to reach the reflector. In addition, you can specify the same information for the return path from the reflector to the client. This return path information is embedded in the probe itself by using the extensions proposed in *Simple TWAMP (STAMP) Extensions for Segment Routing Networks*, draft-ietf-ippm-stamp-srpm, namely the return path TLV and its return segment list sub-TLVs, as appropriate depending on the segment routing type. The TWAMP probes are timestamped in either the Routing Engine or the Packet Forwarding Engine.

To configure this feature for Junos OS Evolved, include the `source-routing` statement at the `[edit services monitoring twamp client control-connection` *`name`* `test-session` *`session-name`*`]` hierarchy level. To configure this feature for Junos OS, include the `source-routing` statement at the `[edit services rpm twamp client control-connection` *`name`* `test-session` *`session-name`*`]` hierarchy level.

Timestamping

Timestamps were originally set only in the Routing Engine. To account for latency in the communication of probe messages, you can offload timestamping of the probe packets to the Packet Forwarding Engine. You can also configure inline timestamping, where the timestamping is done in hardware at the generator or the reflector. Support for this offloading and inline timestamping varies by release, platform, and line card support:

- **Junos OS:** Prior to Junos OS Release 25.4R1, you could not offload timestamping to the Packet Forwarding Engine. You could enable inline timestamping by configuring an `si-` or `sp-` interface. Starting in Junos OS Release 25.4R1, you can offload timestamping to the Packet Forwarding Engine. For those MX line cards that support it, you can also now configure inline timestamping. This inline-timestamping feature also supports Flex Algo and SR-MPLS.

If si- interfaces are configured on an MX router for TWAMP, then this option is the default for timestamping. To debug the si- interface implementation, you need to configure either the none or the pfe-timestamp option.

You configure the offload-type (inline-timestamping | none | pfe-timestamping) statement at one of these hierarchy levels: [edit services rpm twamp client control-connection *name* test-session *name*], [edit services rpm twamp server], or [edit services rpm twamp server light].

- **Junos OS Evolved:** Timestamps are set by the Routing Engine or the Packet Forwarding Engine for IPv4 traffic. For IPv6 traffic, timestamps set by the Routing Engine only. For IPv6 traffic, starting in Junos OS Evolved 22.3R1, we support Packet Forwarding Engine timestamps. Prior to Junos OS Evolved Release 22.3R1, for IPv6 traffic, the offload-type statement at the [edit services monitoring twamp client control-connection *name* test-session *name*] hierarchy level should be configured as none. Starting in Junos OS Evolved Release 22.4R1 for supported devices, you can configure the inline-timestamping option of the offload-type statement to enable timestamps set inline by the hardware. Starting in Junos OS Evolved Release 23.4R1 for servers, the default for the offload-type statement is now pfe-timestamp instead of inline-timestamp. Starting in Junos OS Evolved Release 25.4R1, the inline-timestamping feature also supports Flex Algo and SR-MPLS.

Junos OS Evolved Differences in TWAMP Support

The Junos OS Evolved support for TWAMP is limited to the following:

- IPv4 and IPv6 traffic only for control sessions and test sessions. Starting in Junos OS Evolved Release 21.4R1, IPv6 source and target addresses (except for link-local addresses) are supported for client lists, control connections, and test sessions.
- Probe statistics and history
- Control and test session status
- Test session probe generation and reception, as well as reflection
- Error reporting through system log messages and SNMP traps only
- Unauthenticated mode only

Junos OS Evolved support for TWAMP also includes some features that are not included in Junos OS:

- Starting in Junos OS Evolved Release 23.4R1, we support RFC 8762, *Simple Two-Way Active Measurement Protocol* (STAMP). RFC 8762 standardizes and expands upon the TWAMP Light operational mode, which was defined in Appendix I of RFC 5357, *Two-Way Active Measurement Protocol* (TWAMP). For more information, see "Simple Two-Way Active Measurement Protocol (STAMP) Support" on page 747.

- Starting in Junos OS Evolved Release 24.4R1, we support static route tracking using TWAMP, for those devices that support this feature. See "Static Route Tracking" on page 747 for more information.

Platform-Specific TWAMP Behavior

Use Feature Explorer: Two-Way Active Measurement Protocol to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platform.

Table 129: Platform-Specific TWAMP Behavior

Platform	Difference
ACX Series	• In Junos OS, the ACX710 and ACX5448 Series routers support both reflection and generation. Other ACX Series routers running Junos OS support only reflection, not generation. • In Junos OS Evolved, TWAMP is supported for ACX routers, for both reflection and generation. See "Junos OS Evolved Differences in TWAMP Support" on page 749 for notes about OS differences in TWAMP support. • For Junos OS, TWAMP is configured at the `[edit services rpm twamp]` hierarchy level. For Junos OS Evolved, TWAMP is configured at the `[edit services monitoring twamp]` hierarchy level. • TWAMP control connection traffic always arrives on ACX routers with the listening port set as 862. Because this port number for traffic probes can be modified, probes that arrive with a different port number are not recognized and processed by ACX routers correctly. As a result, TWAMP traffic and host-bound packets are dropped in such a scenario. • For `inline-timestamping` mode, The TWAMP client and server cannot be configured on the same ACX router. • For `inline-timestamping` mode, TWAMP over Layer 3 VPN is not supported. • Prior to Junos OS Evolved Release 23.4R1, if both TWAMP and connectivity fault management (CFM) are configured, the TWAMP client drops the TWAMP probes. Remove the CFM configuration to enable TWAMP. Starting in Junos OS Evolved Release 23.4R1, you can configure CFM on the same device as TWAMP.

Table 129: Platform-Specific TWAMP Behavior *(Continued)*

Platform	Difference
EX Series	Both the control client and session sender (the TWAMP client) reside on the same Juniper Networks router. However, the TWAMP client does not require that the server and the session reflector to be on the same system. Therefore, the Juniper TWAMP client is capable of working with a third-party server implementation.
MX Series	• Both the control client and session sender (the TWAMP client) reside on the same Juniper Networks router. However, the TWAMP client does not require that the server and the session reflector to be on the same system. Therefore, the Juniper TWAMP client is capable of working with a third-party server implementation. • TWAMP is not supported when you enable Next Gen Services on an MX Series router.
PTX Series	• The destination interface `si-x/y/z` attribute, which is meant for enabling inline services, is not supported on PTX Series routers for TWAMP client configurations. • For Junos OS, TWAMP is configured at the `[edit services rpm twamp]` hierarchy level. For Junos OS Evolved, TWAMP is configured at the `[edit services monitoring twamp]` hierarchy level. See "Junos OS Evolved Differences in TWAMP Support" on page 749 for notes about OS differences in TWAMP support.
QFX 10000 Series	Both the control client and session sender (the TWAMP client) reside on the same Juniper Networks router. However, the TWAMP client does not require that the server and the session reflector to be on the same system. Therefore, the Juniper TWAMP client is capable of working with a third-party server implementation.

Table 129: Platform-Specific TWAMP Behavior *(Continued)*

Platform	Difference
QFX5000 Series (Junos OS Evolved)	For Junos OS Evolved, TWAMP is configured at the `[edit services monitoring twamp]` hierarchy level. See "Junos OS Evolved Differences in TWAMP Support" on page 749 for notes about OS differences in TWAMP support.
SRX Series (SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, SRX4100, and SRX4200 devices and vSRX Virtual Firewall instances)	• TWAMP for IPv6 is not supported. • TWAMP server and TWAMP client authentication are not supported. • TWAMP Light is not supported.

For the MX Series, the table below shows the relationship between RPM client and server support, TWAMP client (with the control component) and TWAMP server (with the responder component) support, and the hardware that performs timestamping.

Table 130: TWAMP Feature Support and Hardware for Junos OS, MX Series

TWAMP Feature Support	Routing Engine Timestamp	MS-PIC/MS-DPC Timestamp	MS-MIC/MS-MPC Timestamp	Packet Forwarding Engine (microkernel) Timestamp	Packet Forwarding Engine (LU) Timestamp (`si-` interface)
RPM Client	Yes	Yes	Yes	Yes	No
RPM Server	Yes	Yes	Yes	Yes	No
TWAMP Client	No	No	No	Yes	Yes
TWAMP Server	No	Yes	No	Yes (No responder configuration needed)	Yes

NOTE: Support for the services interfaces (`sp-`, `ms-`, and `si-` interfaces) are all slightly different.

Table 131 on page 754 provides information about MX Series TWAMP and related timestamp support on MPC, MS-MIC/MPC, and inline:

Table 131: Managed TWAMP and related timestamp support, MX Series

Feature	Role	IP Version	Support (Y/N)	Timestamp Inline	Timestamp on MPC (hardware-timestamp)	Timestamp on MPC (si-interface)	Timestamp on MS-MIC/MPC (delegate-probes)
TWAMP	Managed Client	IPv4	Y	N	Y (µsec) 2000 maximum probes	Y (µsec) 2000 maximum probes	N
		IPv6	N	N	N	N	N
	Managed Server	IPv4	Y	N	Y (µsec) 2000 maximum probes	Y (µsec) 2000 maximum probes	N
		IPv6	N	N	N	N	N

Table 132: TWAMP Light and related timestamp support, MX Series

Feature	Role	IP Version	Support (Y/N)	Timestamp Inline	Timestamp on MPC (hardware-timestamp)	Timestamp on MPC (si-interface)	Timestamp on MS-MIC/MPC (delegate-probes)
TWAMP	Light Client	IPv4	Y	Y (µsec) (Starting in Release 25.4R1) 2000 maximum probes	Y (µsec) 2000 maximum probes	Y (µsec) 2000 maximum probes	N
		IPv6	Y	Y (µsec) (Starting in Release 25.4R1) 2000 maximum probes	Y (µsec) 2000 maximum probes	Y (µsec) 2000 maximum probes	N
	Light Server	IPv4	Y	Y (µsec) (Starting in Release 25.4R1) 2000 maximum probes	Y (µsec) 2000 maximum probes	Y (µsec) 2000 maximum probes	N
		IPv6	Y	Y (µsec) (Starting in Release 25.4R1) 2000 maximum probes	Y (µsec) 2000 maximum probes	Y (µsec) 2000 maximum probes	N

RELATED DOCUMENTATION

Example: Configuring TWAMP Client and Server on MX Series Routers

Example: Configuring TWAMP Client and Server for SRX Series Firewalls

Configure TWAMP

Configure TWAMP

SUMMARY

The Two-Way Active Measurement Protocol (TWAMP) defines a standard for measuring IP performance between two devices in a network. For more information on TWAMP, see RFC 5357, *A Two-Way Active Measurement Protocol (TWAMP)*. For more background information on TWAMP, see "Understand Two-Way Active Measurement Protocol" on page 744.

IN THIS SECTION

Understand TWAMP Configuration

Two-Way Active Measurement Protocol (TWAMP) support and configuration varies for hardware platform, physical interfaces, or virtual physical (services) interfaces. Support for RPM is not always an indicator of TWAMP support on a particular combination of platform and line card for Junos OS. The time stamps used in RPM and TWAMP are added in different places, depending on the hardware configuration. For example, different hardware components perform timestamping, either inline in the lookup (LU) chip, Routing Engine (Junos OS Evolved), the microkernel-based timestamping at the host Packet Forwarding Engine, or the line card such as a Multiservices Physical Interface Card (MS-PIC), Multiservices Modular Interface Card (MS-MIC), Multiservices Modular PIC Concentrator (MS-MPC), or Multiservices Dense Port Concentrator (MS-DPC).

Use Feature Explorer: Two-Way Active Measurement Protocol to confirm platform and release support for specific features.

See Understand Two-Way Active Measurement Protocol for Junos OS Evolved differences and notes about your platform.

TWAMP Light Support

TWAMP Light, as defined in Appendix I of RFC 5357, is a stateless version of TWAMP where test parameters are predefined instead of negotiated. All test packets received by the server on a test port are reflected back and forgotten right away.

Support for IPv6 target addresses for TWAMP Light test sessions is introduced in Junos OS Release 21.3R1. For the Junos OS IPv6 TWAMP Light client, you must configure both the `target-address` and the `destination-port` statements at the `[edit services rpm twamp client control-connection *control-client-name* test-session *test-session-name*]` hierarchy level. Support for link-local target addresses for IPv6 TWAMP Light test sessions is introduced starting in Junos OS Release 21.4R1 and in Junos OS Evolved Release 22.3R1, for those devices that support TWAMP Light.

Simple Two-Way Active Measurement Protocol (STAMP) Support

STAMP, as defined in RFC 8762, *Simple Two-Way Active Measurement Protocol* (STAMP), standardizes and expands upon the TWAMP Light operational mode, which was defined in Appendix I of RFC 5357, *Two-Way Active Measurement Protocol* (TWAMP). A STAMP-compliant reflector ensures symmetric payload size (in accordance with RFC 6038) and operates in either stateless or stateful mode, depending on whether the sequence number in the reflected payload is copied from the client frame or generated independently. A stateful reflector can detect in which direction drops have occurred. In previous releases, we supported symmetric payloads and stateless reflection. We now support stateful reflection, full compliance with the STAMP standard, and unidirectional drop values for clients. We support unidirectional drop values not only for STAMP clients, but also for TWAMP-Managed-mode clients. For Junos OS Evolved, STAMP is configured at the [edit services monitoring twamp server light] hierarchy level. Stateful reflection is configured with the `stateful-sequence` statement. For servers, the new default for `offload-type` is now `pfe-timestamp` instead of `inline-timestamp`.

Use Feature Explorer: Simple Two-Way Active Measurement Protocol (STAMP) to confirm platform and release support.

TWAMP Managed Support

For Junos OS, TWAMP is configured at the `[edit services rpm twamp]` hierarchy level. For Junos OS Evolved, TWAMP is configured at the `[edit services monitoring twamp]` hierarchy level.

Use Two-Way Active Measurement Protocol to confirm platform and release support for specific features.

Configure a TWAMP Server

With the exception of physical interfaces, TWAMP server configuration for Junos OS requires the following minimum configuration at the `[edit services rpm twamp]` hierarchy level:

```
server {
    authentication-mode mode;
    client-list list-name {
        address ip-address;
    }
    port 862;
}
```

Starting in Junos OS Release 21.3R1, you no longer need to configure the `authentication-mode` statement. The default mode is now `none`, which means that communications with the server are not authenticated.

- To specify the list of allowed control client hosts that can connect to this server, include the `client-list` statement at the `[edit services rpm twamp server]` hierarchy level. Each value you include must be a Classless Interdomain Routing (CIDR) address (IP address plus mask) that represents a network of allowed hosts. You can include multiple client lists, each of which can contain a maximum of 64 entries. You must configure at least one client address to enable TWAMP.

- ACX Series routers do not support authentication and encryption modes. The value for `authentication-mode` statement at the `[edit services rpm twamp server]` hierarchy level must be set to `none`.

- TWAMP control connection traffic always arrives on ACX routers with the listening port set as 862. Because this port number for traffic probes can be modified, probes that arrive with a different port number are not recognized and processed by ACX routers correctly. As a result, TWAMP traffic and host-bound packets are dropped in such a scenario.

"Configure TWAMP" on page 756 provides information about support for light control of the server.

For Junos OS, you can configure light control for the server (managed control is the default). The Junos OS TWAMP server configuration for light control requires the following minimum configuration at the `[edit services rpm twamp]` hierarchy level:

```
server {
   authentication-mode none;
   light;
   port (862 | 878 | 51000);
}
```

For Junos OS, for a list of restrictions on source addresses, see *source-address (TWAMP)*.

For Junos OS Evolved, you can configure either managed or light control for the server. TWAMP server configuration for managed or light control requires the following minimum configuration at the `[edit services monitoring twamp]` hierarchy level, assuming you use the default port for TWAMP (862):

```
server {
    (managed | light);
}
```

For Junos OS Evolved, you cannot use the following addresses for the client-list source IP address used for probes:

- 0.0.0.0

- 127.0.0.0/8 (loopback)
- 224.0.0.0/4 (multicast)
- 255.255.255.255 (broadcast)

You can configure more than one client, and you can change the TWAMP listening port as long as the change is coordinated with the TWAMP client.

For microkernel-based timestamping in Junos OS, you don't need to configure an `si-` interface. In this case, the TWAMP connection and sessions are established based on the target address and route.

For inline timestamping in Junos OS, you need to configure `si-` or `sp-` services interfaces and the TWAMP server configuration requires the following statements at the `[edit interfaces` *`service-interface-name`*`]` hierarchy level:

```
user@router# show interfaces si-0/0/0
unit 10 {
    rpm twamp-server;
    family inet {
        address 10.10.10.1/24;
    }
}
```

```
user@router# show interfaces sp-0/0/0
unit 10 {
    rpm twamp-server;
    family inet {
        address 10.20.20.1/24;
    }
}
```

NOTE: You cannot configure the TWAMP server on unit 0 of a services interface. If you try, you will receive a configuration error.

(Junos OS only) To configure a TWAMP server on an inline services (`si-`) interface, configure the amount of bandwidth reserved on each Packet Forwarding Engine for tunnel traffic using inline services by including the `bandwidth (1g | 10g)` statement at the `[edit chassis fpc slot-number pic number inline-services]` hierarchy level. Specify the service PIC (`sp-`) logical interface that provides the TWAMP service by including the `twamp-server` statement at the `[edit interfaces sp-`*`fpc/pic/port`*` unit `*`logical-unit-number`*` family inet]` hierarchy level.

The `twamp-server` statement is not required for physical interface TWAMP server configuration.

Many other TWAMP server parameters are optional. See the TWAMP `server` configuration statements for details.

Configure a TWAMP Client

For Junos OS, to configure the TWAMP client service, include the `client` statement and related parameters at the `[edit services rpm twamp]` hierarchy level. For Junos OS Evolved, include the `client` statement and related options at the `[edit services monitoring twamp]` hierarchy level.

There are many options available for TWAMP client configuration. See the configuration statement topics and examples for details.

For microkernel-based timestamping in Junos OS, you don't need to configure an `si-` interface. In this case, the TWAMP connection and sessions are established based on the target address and route.

For inline timestamping in Junos OS, the `si-` interfaces are virtual physical interfaces that respond as a TWAMP server. However, you can also configure services interfaces to act as the TWAMP client, which performs the TWAMP controller role.

(Junos OS only) To configure a services interface as a TWAMP client, you configure the service parameters and the service interface as a TWAMP client.

To configure the TWAMP client services interface, include the `rpm twamp-client` statement at the `[edit interfaces si-interface-name]` hierarchy level:

```
user@router# show interfaces si-0/0/0
unit 0 {
    family inet;
}
unit 10 {
    rpm twamp-client;
    family inet {
        address 10.30.30.1/24
    }
}
```

NOTE: You cannot configure the TWAMP client on unit 0 of a service interface. If you try, you will receive a configuration error.

SEE ALSO

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
25.4R1-EVO	We added support for Flex Algo and SR-MPLS to our inline-timestamping feature.
25.4R1	We support offloading the timestamping to the Packet Forwarding Engine. Also, for those MX line cards that support it, we support inline timestamping, where the timestamping is done in hardware at the generator or the reflector. The inline-timestamping feature also supports Flex Algo and SR-MPLS. You configure the `offload-type` statement on either the server or client.
23.4R1-EVO	We support RFC 8762, *Simple Two-Way Active Measurement Protocol* (STAMP). RFC 8762 standardizes and expands upon the TWAMP Light operational mode, which was defined in Appendix I of RFC 5357, *Two-Way Active Measurement Protocol* (TWAMP). A STAMP-compliant reflector ensures symmetric payload size (in accordance with RFC 6038) and operates in either stateless or stateful mode, depending on whether the sequence number in the reflected payload is copied from the client frame or generated independently. A stateful reflector can detect in which direction drops have occurred. In previous releases, we supported symmetric payloads and stateless reflection. With this release, we support stateful reflection, full compliance with the STAMP standard, and unidirectional drop values for clients. We support unidirectional drop values not only for STAMP clients, but also for TWAMP-Managed-mode clients. For servers, the default for the `offload-type` statement is now `pfe-timestamp` instead of `inline-timestamp`.
22.4R1-EVO	We support IPv6 source and target addresses (except for link-local addresses) for client lists, control connections, and test sessions.
22.4R1-EVO	You can configure SNMP traps for TWAMP.
22.4R1-EVO	We support inline timestamping, where the timestamping is done in hardware at the generator or the reflector.

22.3R1-EVO	For TWAMP Light test sessions, you can specify IPv6 link-local addresses for target addresses.
21.4R1-EVO	IPv6 source and target addresses (except for link-local addresses) are supported for client lists, control connections, and test sessions.
21.4R1	For TWAMP Light test sessions, you can specify IPv6 link-local addresses for target addresses, and can configure IPv6 addresses for source addresses that correspond to target addresses configured with IPv6 link-local addresses.
21.3R1	IPv6 target addresses for TWAMP Light test sessions are supported.
21.3R1	You no longer have to configure the `authentication-mode` statement for the TWAMP server. The default mode is `none`.
21.1R1	We support the Two-Way Active Measurement Protocol (TWAMP) Light client and server, as defined in Appendix I of RFC 5357, for IPv4 target addresses. TWAMP Light is a stateless version of TWAMP, where test parameters are predefined instead of negotiated. All test packets received by the server on a test port are reflected back and forgotten right away.

Example: Configuring TWAMP Client and Server on MX Series Routers

IN THIS SECTION

This example shows how to configure the TWAMP client and server and contains the following sections.

Requirements

This example uses the following hardware and software components:

- MX Series routers.

- Junos OS Release 15.1 or later.

Overview

This example explains the Two-Way Active Measurement Protocol (TWAMP). TWAMP is an open protocol for measuring network performance between any two devices supporting the TWAMP protocol. The TWAMP-Control protocol is used to set up performance measurement sessions. The TWAMP-Test protocol is used to send and receive performance measurement probes.

The TWAMP architecture is composed of the following entities that are responsible for starting a monitoring session and exchanging packets:

- The control client initiates all requested test sessions with a start sessions message, and the server acknowledges. When necessary, the control client sends a message to stop all test sessions.
- The session sender and the session reflector exchange test packets according to the TWAMP-Test protocol for each active session. On receiving a TWAMP-Test packet, the session reflector only reflects a measurement packet and does not collect packet statistics in TWAMP.

The TWAMP server is an end system that manages one or more TWAMP sessions and is also capable of configuring per-session ports. The server listens on the TCP port. The session reflector and server make up the TWAMP responder in an IP service-level agreement operation.

For 15.1, both the control client and session sender would be residing on the same Juniper router. The client design does not mandate the server and the session reflector to be on the same system. Hence the Juniper TWAMP client will also be capable of working with a third-party server implementation.

Configuration for TWAMP client

IN THIS SECTION

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of SG1 router.

Configuring Chassis

```
set chassis fpc 4 pic 1 inline-services bandwidth 1g
```

Configuring Interfaces

```
set interfaces si-4/1/0 unit 0 family inet
set interfaces si-4/1/0 unit 10 rpm twamp-client
set interfaces si-4/1/0 unit 10 family inet address 10.60.60.1/32
```

Configuring Services

```
set services rpm twamp client control-connection c1 destination-interface si-4/1/0.10
set services rpm twamp client control-connection c1 history-size 500
set services rpm twamp client control-connection c1 target-address 10.70.70.1
set services rpm twamp client control-connection c1 test-count 1
set services rpm twamp client control-connection c1 test-interval 1
set services rpm twamp client control-connection c1 traps test-iteration-done
set services rpm twamp client control-connection c1 traps control-connection-closed
set services rpm twamp client control-connection c1 test-session t1 target-address 10.70.70.1
set services rpm twamp client control-connection c1 test-session t1 data-fill-with-zeros
set services rpm twamp client control-connection c1 test-session t1 data-size 1400
set services rpm twamp client control-connection c1 test-session t1 probe-count 55
set services rpm twamp client control-connection c1 test-session t1 probe-interval 1
```

Configuring TWAMP client

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

1. Configure the chassis.

```
[edit chassis]
user@router1# set fpc 4 pic 1 inline-services bandwidth 1g
```

2. Configure the interfaces.

```
[edit interfaces]
user@router1# set si-4/1/0 unit 0 family inet
user@router1# set si-4/1/0 unit 10 rpm twamp-client
user@router1# set si-4/1/0 unit 10 family inet address 10.60.60.1/32
```

3. Configure the services.

```
[edit services]
user@router1# set rpm twamp client control-connection c1 destination-interface si-4/1/0.10
user@router1# set rpm twamp client control-connection c1 history-size 500
user@router1# set rpm twamp client control-connection c1 target-address 10.70.70.1
user@router1# set rpm twamp client control-connection c1 test-count 1
user@router1# set rpm twamp client control-connection c1 test-interval 1
user@router1# set rpm twamp client control-connection c1 traps test-iteration-done
user@router1# set rpm twamp client control-connection c1 traps control-connection-closed
user@router1# set rpm twamp client control-connection c1 test-session t1 target-address
10.70.70.1
user@router1# set rpm twamp client control-connection c1 test-session t1 data-fill-with-zeros
user@router1# set rpm twamp client control-connection c1 test-session t1 data-size 1400
user@router1# set rpm twamp client control-connection c1 test-session t1 probe-count 55
user@router1# set rpm twamp client control-connection c1 test-session t1 probe-interval 1
```

Results

From the configuration mode of Router 1, confirm your configuration by entering the `show chassis`, `show interfaces`, and `show services rpm twamp` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@router1# show chassis
fpc 4 {
    pic 1 {
        inline-services {
            bandwidth 1g;
        }
```

```
    }
}
```

```
user@router1# show interfaces
    si-4/1/0 {
        unit 0 {
            family inet;
        }
        unit 10 {
            rpm twamp-client;
            family inet {
                address 10.60.60.1/32;
            }
        }
    }
```

```
user@router1# show services rpm twamp
client {
    control-connection c1 {
        destination-interface si-4/1/0.10;
        history-size 500;
        target-address 10.70.70.1;
        test-count 1;
        test-interval 1;
        traps {
            test-iteration-done;
            control-connection-closed;
        }
        test-session t1 {
            target-address 10.70.70.1;
            data-fill-with-zeros;
            data-size 1400;
            probe-count 55;
            probe-interval 1;
        }
    }
}
```

Configuration for TWAMP server

IN THIS SECTION

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of SG1 router.

Configuring Chassis

```
set chassis fpc 2 pic 1 inline-services bandwidth 1g
```

Configuring Interfaces

```
set interfaces si-2/1/0 unit 0 family inet
set interfaces si-2/1/0 unit 10 rpm twamp-server
set interfaces si-2/1/0 unit 10 family inet address 10.70.70.1/32
```

Configuring Services

```
set services rpm twamp server authentication-mode none
set services rpm twamp server port 862
set services rpm twamp server client-list Client1 address 10.60.60.1/32
```

Configuring TWAMP server

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

1. Configure the chassis.

```
[edit chassis]
user@router1# set fpc 2 pic 1 inline-services bandwidth 1g
```

2. Configure the interfaces.

```
[edit interfaces]
user@router1#set si-2/1/0 unit 0 family inet
user@router1#set si-2/1/0 unit 10 rpm twamp-server
user@router1#set si-2/1/0 unit 10 family inet address 10.70.70.1/32
```

3. Configure the services.

```
[edit services]
user@router1# set rpm twamp server authentication-mode none
user@router1# set rpm twamp server port 862
user@router1# set rpm twamp server client-list Client1 address 10.60.60.1/32
```

Results

From the configuration mode of Router 1, confirm your configuration by entering the `show chassis`, `show interfaces`, and `show services rpm twamp server` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@router1# show chassis
fpc 2 {
    pic 1 {
        inline-services {
            bandwidth 1g;
        }
    }
}
```

```
user@router1# show interfaces
si-2/1/0 {
    unit 0 {
```

```
        family inet;
    }
    unit 10 {
        rpm twamp-server;
        family inet {
            address 10.70.70.1/32;
        }
    }
}
```

```
user@router1# show services rpm twamp server
authentication-mode none;
    port 862;
    client-list Client1 {
    address {
        10.60.60.1/32;
    }
}
```

Verification

IN THIS SECTION

Verifying TWAMP server sessions

Purpose

Verify that the TWAMP server sessions are established.

Action

From operational mode, enter the `show services rpm twamp server session` command.

```
user@router1> show services rpm twamp server session
    Session  Connection      Sender           Sender    Reflector          Reflector
         ID          ID      address            port      address               port
          4          44     10.1.1.1           12345  192.168.219.203        890
         78          44     10.22.1.55           345  10.22.2.2              89022
        234         423     192.168.219.203     2345  10.2.22.2              3333
          5         423     10.4.1.1           82345  10.2.2.2               45909
          1         423     192.168.1.1          645  10.32.2.2              2394
```

Verifying TWAMP client sessions

Purpose

Verify that the TWAMP client sessions are established.

Action

From operational mode, enter the `show services rpm twamp client session` command.

```
user@router1> show services rpm twamp client session

Connection  Session   Sender         Sender    Reflector     Reflector
      Name     Name   address          port      address          port
        c2       t1   10.60.60.1      10008   10.70.70.1         10008
```

RELATED DOCUMENTATION

request services rpm twamp

Example: Configuring TWAMP Client and Server for SRX Series Firewalls

IN THIS SECTION

This example shows how to configure the Two-Way Active Measurement Protocol (TWAMP) client and TWAMP server.

NOTE: Our content testing team has validated and updated this example.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall.
- Junos OS Release 18.1R1 and later releases.
  - Updated and revalidated using vMX on Junos OS Release 22.2R1.

Before you begin configuring TWAMP client and TWAMP server, ensure that you have read "Understand Two-Way Active Measurement Protocol" on page 744 to understand how this task fits into the overall configuration process.

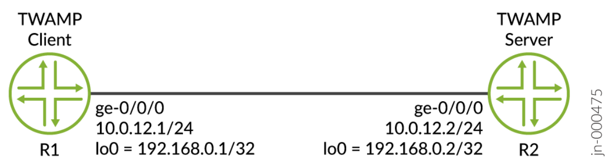
Overview

The TWAMP is an open protocol for measuring network performance between any two devices in a network that supports the TWAMP protocol. The TWAMP consists of TWAMP-Control protocol and TWAMP-Test protocol. The TWAMP-Control protocol is used to initiate, start and stop the test sessions between the control client. The TWAMP-Test protocol used to exchange the test packets between the session sender and the session reflector.

Figure 66 on page 772 shows the TWAMP architecture composed of the following entities that are responsible for starting a monitoring session and exchanging packets:

- The control client initiates all requested test sessions with a start sessions message, and the TWAMP server acknowledges. When necessary, the control client sends a message to stop all test sessions.
- The session sender and the session reflector exchange test packets according to the TWAMP-Test protocol for each active session. On receiving a TWAMP-Test packet, the session reflector reflects a measurement packet and does not collect any packet statistics in TWAMP.

Figure 66: Configuring TWAMP Client and TWAMP Server



The TWAMP server is an end system that manages one or more TWAMP sessions and capable of configuring per-session ports. The TWAMP server listens to the TCP port. The session reflector and TWAMP server make up the TWAMP responder in an IP service-level agreement operation.

For Junos OS Release 18.1R1, both the control client and session sender resides on the same device. The client design does not mandate the TWAMP server and the session reflector to be on the same system. Hence, the Juniper TWAMP client is also capable of working with a third-party server implementation.

Configuring the TWAMP Client for SRX Series Firewalls

CLI Quick Configuration

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

```
set system host-name R1
set services rpm twamp client control-connection c1 target-address 10.0.12.2
set services rpm twamp client control-connection c1 test-session t1 target-address 10.0.12.2
set services rpm twamp client control-connection c1 test-session t1 probe-count 2000
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/0.0
```

```
set interfaces ge-0/0/0 unit 0 description "To Server R2"
set interfaces ge-0/0/0 unit 0 family inet address 10.0.12.1/24
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the TWAMP Client:

1. Configure the client device host name as R1.

   ```
   [edit system]
   user@R1# set host-name R1
   ```

2. Configure Device R1 interfaces.

   ```
   [edit interfaces]
   user@R1# set ge-0/0/0 unit 0 description "To Server R2"
   user@R1# set ge-0/0/0 unit 0 family inet address 10.0.12.1/24
   user@R1# set lo0 unit 0 family inet address 192.168.0.1/32
   ```

3. Enable traffic flow and system services to run on Device R1, which is otherwise dropped by default.

   ```
   [edit security zones]
   user@R1# set security-zone trust host-inbound-traffic system-services all
   user@R1# set security-zone trust host-inbound-traffic protocols all
   user@R1# set security-zone trust interfaces ge-0/0/0.0
   ```

4. Configure the control session from Device R1 to Device R2.

   ```
   [edit services]
   user@R1# set rpm twamp client control-connection c1 target-address 10.0.12.2
   ```

5. Configure the test session from Device R1 to Device R2 for collecting probe results.

```
[edit services]
user@R1# set rpm twamp client control-connection c1 test-session t1 target-address 10.0.12.2
user@R1# set rpm twamp client control-connection c1 test-session t1 probe-count 2000
```

Results

From the configuration mode on Device R1, confirm your configuration by entering the `show | no-more` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@R1# show | no-more
system {
    host-name R1;
}
services {
    rpm {
        twamp {
            client {
                control-connection c1 {
                    target-address 10.0.12.2;
                    test-session t1 {
                        target-address 10.0.12.2;
                        probe-count 2000;
                    }
                }
            }
        }
    }
}
security {
    policies {
        default-policy {
            permit-all;
        }
    }
    zones {
        security-zone trust {
            host-inbound-traffic {
```

```
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                ge-0/0/0.0;
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            description "To Server R2";
            family inet {
                address 10.0.12.1/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.1/32;
            }
        }
    }
}
```

Configuring the TWAMP Server for SRX Series Firewalls

CLI Quick Configuration

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

```
set system host-name R2
```

```
set services rpm twamp server authentication-mode none
set services rpm twamp server client-list client1 address 10.0.12.1/24
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 description "To Client R1"
set interfaces ge-0/0/0 unit 0 family inet address 10.0.12.2/24
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the TWAMP Server:

1. Configure the server device host name as R2.

   ```
   [edit system]
   user@R2# set host-name R2
   ```

2. Configure Device R2 interfaces.

   ```
   [edit interfaces]
   user@R2# set ge-0/0/0 unit 0 description "To Client R1"
   user@R2# set ge-0/0/0 unit 0 family inet address 10.0.12.2/24
   user@R2# set lo0 unit 0 family inet address 192.168.0.2/32
   ```

3. Enable traffic flow and system services to run on Device R2, which is otherwise dropped by default.

   ```
   [edit security zones]
   user@R2# set security-zone trust host-inbound-traffic system-services all
   user@R2# set security-zone trust host-inbound-traffic protocols all
   user@R2# set security-zone trust interfaces ge-0/0/0.0
   ```

4. Configure the client attributes for Device R2 to connect with Device R1.

```
[edit services]
user@R2# set rpm twamp server authentication-mode none
user@R2# set rpm twamp server client-list client1 address 10.0.12.1/24
```

Results

From the configuration mode on R2, confirm your configuration by entering the `show | no-more` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@R2# show | no-more
system {
    host-name R2;
}
services {
    rpm {
        twamp {
            server {
                authentication-mode none;
                client-list client1 {
                    address {
                        10.0.12.1/24;
                    }
                }
            }
        }
    }
}
security {
    policies {
        default-policy {
            permit-all;
        }
    }
    zones {
        security-zone trust {
            host-inbound-traffic {
                system-services {
```

```
                    all;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                ge-0/0/0.0;
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            description "To Client R1";
            family inet {
                address 10.0.12.2/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.2/32;
            }
        }
    }
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying TWAMP Client Sessions

Purpose

Verify that the TWAMP client sessions are established on Device R1.

Action

From operational mode, enter the `show services rpm twamp client session` command.

```
user@R1>show services rpm twamp client session

Connection      Session         Sender          Sender Reflector        Reflector
Name            Name            address         port   address          port
c1              t1              10.0.12.1        10010 10.0.12.2         10010
```

Meaning

The configured control and test sessions (c1 and t1, respectively) are established on Device R1.

Verifying TWAMP Server Sessions

Purpose

Verify that the TWAMP server sessions are established on Device R2.

Action

From operational mode, enter the `show services rpm twamp server session` command.

```
user@R2>show services rpm twamp server session
Session  Connection  Sender           Sender  Reflector        Reflector  Session    Auth
ID       ID          address          port    address          port       state      mode
     11           2  10.0.12.1         10010  10.0.12.2            10010  Active
Unauthenticated
```

Meaning

The server session on Device R2 is active with Device R1 as the sender and Device R2 as the reflector.

Verifying Test Session Results

Purpose

Verify that the TWAMP test sessions on Device R1.

Action

From operational mode, enter the `show services rpm twamp client probe-results` command.

```
user@R1> show services rpm twamp client probe-results
Owner: c1, Test: t1
    server-address: 10.0.12.2, server-port: 862, Client address: 10.0.12.1, Client port: 60732
    TWAMP-Server-Status: Connected, Number-Of-Retries-With-TWAMP-Server: 38
    Reflector address: 10.0.12.2, Reflector port: 10011, Sender address: 10.0.12.1, sender-port:
10011
    Test size: 2000 probes
    Probe results:
      Response received
      Probe sent time: Fri Nov 25 03:18:34 2022
      Probe rcvd/timeout time: Fri Nov 25 03:18:34 2022
      Rtt: 718 usec, Ingress time: 134 usec, Egress time: 584 usec, Egress jitter: 48 usec,
Ingress jitter: 15 usec,
      Round trip jitter: 63 usec
      Egress interarrival jitter: 58 usec, Ingress interarrival jitter: 40 usec, Round trip
interarrival jitter: 80 usec

...(output truncated for brevity)...
```

Meaning

The probe-results of the TWAMP test session is generated. This shows that the client-server connection is established successfully.

Understanding TWAMP Auto-Restart

IN THIS SECTION

After a network outage or a configuration change, when the Two-Way Active Management Protocol (TWAMP) client goes down, you have to manually start the TWAMP session by using `request services rpm twamp start client` command. Starting in Junos OS Release 19.1R1, the TWAMP client restarts automatically without any manual intervention.

The automatic restart of the TWAMP session enables the TWAMP client to initiate the TCP control connection and UDP test sessions automatically during the following scenarios:

- Immediately after the TWAMP client configuration is committed.
- After the remote operation daemon (rmopd) is started with the valid TWAMP client configuration presence.
- After the TWAMP client configuration is activated.
- Immediately after the TWAMP server is reachable from the TWAMP client, based on the `test-interval`.

When the network fails or the TWAMP server becomes unreachable for any reason, the, TWAMP client tries to reconnect to the TWAMP server after every `test-interval` value until it is successful. However, for the client to reconnect to the TWAMP server automatically, the *test-count* value in the `set rpm twamp client control-connection` *`test-count`* command must be 0. At the TWAMP server side, the default value of *`max-connection-duration`* in the `set rpm twamp server max-connection-duration` must also be 0. Thereby, you can retain the connection until it is cleared.

NOTE: Starting in Junos OS Release 19.1R1, the default value of *test-count* at the TWAMP client and *max-connection-duration* at the TWAMP server is 0.

After you configure and commit a TWAMP test, the client runs tests indefinitely—that is, it continues to send probes after the configured test interval even after a test is completed, and even if there is a network or server failure. You can stop the automatic running of tests by changing the value of the *`test-count`* option to a nonzero value. If you do that, the automatic restart feature is disabled, and you need to manually start the TWAMP client for it to establish connection with the server and start test sessions.

You can maintain and view the statistics related to the previous probes sent during server unavailability. You can Use the `set services rpm twamp client control-connection c1 persistent-results` command to preserve and display the test results after the network recovers or when the TWAMP server is again reachable.

Benefits

- You do not need to restart the TWAMP session manually after the client goes down as a result of a network outage or configuration change.
- You do not need to run an event script to restart TWAMP session from client side.

TCP Keepalive Support for TWAMP Client and Server

Keepalive probes can assert client (peers) when another peer becomes unreachable. If the problem is in the network between two peers, the keepalive action is to wait for some time and then retry sending the keepalive packet before marking the connection as broken.

When the keepalive timer for a TCP connection reaches zero, TCP client sends its peer a keepalive probe packet with no data in it and with the ACK flag turned on. The client receives a reply from the remote host with no data and with the ACK flag set. If the client receives a reply to its keepalive probe, the client can assert that the connection is still up and running. If the peer does not reply to the keepalive probe, you can assert that the connection cannot be considered valid and then take corrective action.

In Junos OS, to detect the TWAMP control connection failures at TWAMP client and TWAMP servers, you need to configure the following parameters:

- `tcp-keepcnt`—Number of unacknowledged probes to send before considering the connection dead and notifying the application layer.
- `tcp-keepidle`—Time interval between the last data packet sent and the first keepalive probe sent.
- `tcp-keepintvl`—Time interval between successive keepalive probes.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.1R1	Starting in Junos OS Release 19.1R1, the TWAMP client restarts automatically without any manual intervention.

RELATED DOCUMENTATION

tcp-keepcnt

tcp-keepintvl

tcp-keepidle

Configuring TWAMP Client and TWAMP Server to Reconnect Automatically After TWAMP Server Unavailability | 783

Configuring TWAMP Client and TWAMP Server to Reconnect Automatically After TWAMP Server Unavailability

You can run TWAMP client automatically without any manual intervention during network failures or configuration changes. In case of a network outage or connection loss between a TWAMP client and TWAMP server, all the affected TWAMP TCP control connections and UDP test-sessions are lost. At each test-interval, the TWAMP client continues to send the control packets to re-establish connectivity with TWAMP server till is successful. All the statistics will be maintained during that network failure.

This procedure is for Junos OS only. To configure the TWAMP client:

1. Configure the interfaces.

```
[edit interfaces]
user@router1# set si-2/2/0 unit 0 family inet
user@router1# set si-2/2/0 unit 10 rpm twamp-client
user@router1# set si-2/2/0 unit 10 family inet address 192.168.20.1/32
```

2. Configure the chassis.

```
[edit chassis]
user@router1# set fpc 2 pic 2 inline-services bandwidth 1g
```

3. Configure the services.

```
[edit services]
user@router1# set rpm twamp client control-connection c1 destination-interface si-2/2/0.10
user@router1# set rpm twamp client control-connection c1 persistent-results
user@router1# set rpm twamp client control-connection c1 history-size 500
user@router1# set rpm twamp client control-connection c1 routing instance IN
user@router1# set rpm twamp client control-connection c1 target-address 192.0.2.2
```

```
user@router1# set rpm twamp client control-connection c1 tcp-keepidle 20
user@router1# set rpm twamp client control-connection c1 tcp-keepintvl 4
user@router1# set rpm twamp client control-connection c1 tcp-keepcnt 10
user@router1# set rpm twamp client control-connection c1 test-interval 4
user@router1# set rpm twamp client control-connection c1 traps control-connection-closed
user@router1# set rpm twamp client control-connection c1 test-session t1 target-address
192.0.2.2
user@router1# set rpm twamp client control-connection c1 test-session t1 data-fill-with-zeros
user@router1# set rpm twamp client control-connection c1 test-session t1 data-size 1400
user@router1# set rpm twamp client control-connection c1 test-session t1 probe-count 20
user@router1# set rpm twamp client control-connection c1 test-session t1 probe-interval 1
user@router1# set rpm twamp client control-connection c1 test-session t2 target-address
192.0.2.2
user@router1# set rpm twamp client control-connection c1 test-session t1 data-fill-with-zeros
user@router1# set rpm twamp client control-connection c1 test-session t1 data-size 1400
user@router1# set rpm twamp client control-connection c1 test-session t1 probe-count 20
user@router1# set rpm twamp client control-connection c1 test-session t1 probe-interval 1
user@router1# set rpm twamp client control-connection c1 test-session t1 thresholds total-
loss 10
user@router1# set rpm twamp client control-connection c1 test-session t1 thresholds jitter-
gress 20
user@router1# set rpm twamp client control-connection c1 test-session t2 target-address
192.0.3.1
user@router1# set rpm twamp client control-connection c1 test-session t2 data-fill-with-zeros
user@router1# set rpm twamp client control-connection c1 test-session t2 data-size 1400
user@router1# set rpm twamp client control-connection c1 test-session t2 probe-count 15
user@router1# set rpm twamp client control-connection c1 test-session t2 probe-interval 1
user@router1# set rpm twamp client control-connection c1 test-session t2 thresholds total-
loss 10
user@router1# set rpm twamp client control-connection c1 test-session t2 thresholds jitter-
gress 20
```

To configure the TWAMP server:

1. Configure the interfaces.

```
[edit interfaces]
user@router1# set si-1/1/0 unit 30 family inet
user@router1# set si-1/1/0 unit 30 rpm twamp-server
user@router1# set si-1/1/0 unit 30 family inet address 192.02.2/24
```

2. Configure the chassis.

```
[edit chassis]
user@router1# set fpc 1 pic 1 inline-services bandwidth 1g
```

3. Configure the services.

```
[edit services]
user@router1# set rpm twamp server tcp-keepidle 200
user@router1# set rpm twamp server tcp-keepintvl 20
user@router1# set rpm twamp server tcp-keepcnt 210
user@router1# set rpm twamp server authentication-mode none
user@router1# set rpm twamp server server-inactivity-timeout 5
user@router1# set rpm twamp server reflector-inactivity-timeout 15
user@router1# set rpm twamp server max-connection-duration 0
user@router1# set rpm twamp server maximum-sessions 100
user@router1# set rpm twamp server maximum-sessions-per-connection 50
user@router1# set rpm twamp server maximum-connections 500
user@router1# set rpm twamp server maximum-connections-per-client 500
user@router1# set rpm twamp server port 862
user@router1# set rpm twamp server client-list Client1 address 192.168.20.1/24
```

When the TWAMP server is reachable, the output for Junos OS is as follows. The `TWAMP-Server-Status` is `Connected` and the `Number-Of-Retries-With-TWAMP-Server` is 1

```
user@router1> show services rpm twamp client probe-results | no-more
Jan 11 11:43:42
    Owner: c1, Test: t1
    server-address: 192.0.2.2, server-port: 862, Client address: 192.168.20.1, Client port: 58991
    TWAMP-Server-Status: Connected, Number-Of-Retries-With-TWAMP-Server: 1
    Routing Instance Name: IN
    Destination interface name: si-2/2/0.10
    Test size: 20 probes
    Probe results:
      Response received
      Probe sent time: Fri Jan 11 11:43:41 2019
      Probe rcvd/timeout time: Fri Jan 11 11:43:41 2019
      Rtt: 57 usec, Egress jitter: 1 usec, Ingress jitter: -1 usec, Round trip jitter: 0 usec
      Egress interarrival jitter: 43 usec, Ingress interarrival jitter: 43 usec, Round trip
interarrival jitter: 1 usec
```

```
    Results over current test:

.......
........
    Owner: c1, Test: t2
    server-address: 192.0.2.2, server-port: 862, Client address: 192.168.20.1, Client port: 58991
    TWAMP-Server-Status: Connected, Number-Of-Retries-With-TWAMP-Server: 1
    Routing Instance Name: IN
    Destination interface name: si-2/2/0.10
    Test size: 15 probes
   Probe results:
      Response received
      Probe sent time: Fri Jan 11 11:43:36 2019
      Probe rcvd/timeout time: Fri Jan 11 11:43:36 2019
      Rtt: 58 usec, Egress jitter: 1 usec, Ingress jitter: -1 usec, Round trip jitter: 0 usec
      Egress interarrival jitter: 28 usec, Ingress interarrival jitter: 28 usec, Round trip
interarrival jitter: 0 usec
    Results over current test:
      Probes sent: 15, Probes received: 15, Loss percentage: 0.000000
      Measurement: Round trip time
        Samples: 15, Minimum: 57 usec, Maximum: 59 usec, Average: 58 usec, Peak to peak: 2 usec,
Stddev: 1 usec, Sum: 866 usec
      Measurement: Positive egress jitter

........
    Measurement: Round trip time
        Samples: 105, Minimum: 57 usec, Maximum: 59 usec, Average: 58 usec, Peak to peak: 2
usec, Stddev: 1 usec, Sum: 6062 usec
      Measurement: Positive egress jitter
        Samples: 77, Minimum: 0 usec, Maximum: 398 usec, Average: 12 usec, Peak to peak: 398
usec, Stddev: 63 usec, Sum: 925 usec
      Measurement: Negative egress jitter
        Samples: 18, Minimum: 16 usec, Maximum: 431 usec, Average: 69 usec, Peak to peak: 415
usec, Stddev: 91 usec, Sum: 1248 usec
      Measurement: Positive ingress jitter
        Samples: 19, Minimum: 0 usec, Maximum: 431 usec, Average: 66 usec, Peak to peak: 431
usec, Stddev: 90 usec, Sum: 1249 usec
      Measurement: Negative ingress jitter
        Samples: 76, Minimum: 1 usec, Maximum: 397 usec, Average: 12 usec, Peak to peak: 396
usec, Stddev: 63 usec, Sum: 922 usec
      Measurement: Positive round trip jitter
        Samples: 79, Minimum: 0 usec, Maximum: 1 usec, Average: 0 usec, Peak to peak: 1 usec,
Stddev: 0 usec, Sum: 26 usec
```

```
      Measurement: Negative round trip jitter
        Samples: 25, Minimum: 1 usec, Maximum: 1 usec, Average: 1 usec, Peak to peak: 0 usec, 
Stddev: 0 usec, Sum: 25 usec
```

After the server is deactivated using the command `deactivate interfaces si-1/1/0 unit 30` , the output is as follows for Junos OS. The `TWAMP-Server-Status` is `Not Connected` and the `Number-Of-Retries-With-TWAMP-Server` is 12:

```
user@router1> show services rpm twamp client probe-results control-connection c1 | no-more
Jan 11 11:48:24
    Owner: c1, Test: t1
    server-address: 192.0.2.2, server-port: 862, Client address: 192.168.20.1, Client port: 58991
    TWAMP-Server-Status: Not Connected, Number-Of-Retries-With-TWAMP-Server: 12
    Reflector address: 192.0.2.2, Reflector port: 14779, Sender address: 192.168.20.1, sender-
port: 14779
    Routing Instance Name: IN
    Destination interface name: si-2/2/0.10
    Test size: 20 probes
    Probe results:
      Response received
      Probe sent time: Fri Jan 11 11:45:38 2019
      Probe rcvd/timeout time: Fri Jan 11 11:45:38 2019
      Rtt: 55 usec, Egress jitter: -17 usec, Ingress jitter: 18 usec, Round trip jitter: 1 usec
      Egress interarrival jitter: 37 usec, Ingress interarrival jitter: 37 usec, Round trip 
interarrival jitter: 0 usec
    Results over current test:
      Probes sent: 10, Probes received: 10, Loss percentage: 0.000000
      Measurement: Round trip time

.........
   Samples: 17, Minimum: 0 usec, Maximum: 3 usec, Average: 0 usec, Peak to peak: 3 usec, Stddev: 
1 usec, Sum: 4 usec
      Measurement: Negative round trip jitter
        Samples: 3, Minimum: 1 usec, Maximum: 3 usec, Average: 2 usec, Peak to peak: 2 usec, 
Stddev: 1 usec, Sum: 5 usec
    Results over all tests:
      Probes sent: 210, Probes received: 210, Loss percentage: 0.000000



........

TWAMP-Server-Status: Not Connected, Number-Of-Retries-With-TWAMP-Server: 12
```

```
    Reflector address: 192.0.2.2, Reflector port: 14778, Sender address: 192.168.20.1, sender-
port: 14778
    Routing Instance Name: IN
    Destination interface name: si-2/2/0.10
    Test size: 15 probes
    Probe results:
      Response received
      Probe sent time: Fri Jan 11 11:45:38 2019
      Probe rcvd/timeout time: Fri Jan 11 11:45:38 2019
      Rtt: 58 usec, Egress jitter: -18 usec, Ingress jitter: 19 usec, Round trip jitter: 0 usec

..........
    Results over all tests:
      Probes sent: 160, Probes received: 160, Loss percentage: 0.000000
      Measurement: Round trip time
        Samples: 160, Minimum: 57 usec, Maximum: 59 usec, Average: 58 usec, Peak to peak: 2
usec, Stddev: 1 usec, Sum: 9232 usec
      Measurement: Positive egress jitter
        Samples: 119, Minimum: 0 usec, Maximum: 398 usec, Average: 12 usec, Peak to peak: 398
usec, Stddev: 62 usec, Sum: 1398 usec
      Measurement: Negative egress jitter
        Samples: 27, Minimum: 16 usec, Maximum: 431 usec, Average: 64 usec, Peak to peak: 415
usec, Stddev: 76 usec, Sum: 1723 usec
      Measurement: Positive ingress jitter
        Samples: 28, Minimum: 0 usec, Maximum: 431 usec, Average: 62 usec, Peak to peak: 431
usec, Stddev: 76 usec, Sum: 1727 usec
      Measurement: Negative ingress jitter
        Samples: 118, Minimum: 1 usec, Maximum: 397 usec, Average: 12 usec, Peak to peak: 396
usec, Stddev: 62 usec, Sum: 1400 usec
      Measurement: Positive round trip jitter
        Samples: 120, Minimum: 0 usec, Maximum: 1 usec, Average: 0 usec, Peak to peak: 1 usec,
Stddev: 0 usec, Sum: 39 usec
      Measurement: Negative round trip jitter
        Samples: 39, Minimum: 1 usec, Maximum: 1 usec, Average: 1 usec, Peak to peak: 0 usec,
Stddev: 0 usec, Sum: 39 usec
```

After activating the server using the `activate interfaces si-1/1/0 unit 30` command the output is as follows for Junos OS. The `TWAMP-Server-Status` is `Connected` and the `Number-Of-Retries-With-TWAMP-Server` is 12.

```
user@router1> show services rpm twamp client probe-results control-connection c1 | no-more
Jan 11 11:48:50
    Owner: c1, Test: t1
```

```
    server-address: 192.0.2.2, server-port: 862, Client address: 192.168.20.1, Client port: 58991
    TWAMP-Server-Status: Connected, Number-Of-Retries-With-TWAMP-Server: 12
    Reflector address: 192.0.2.2, Reflector port: 14963, Sender address: 192.168.20.1, sender-
port: 14963
    Routing Instance Name: IN
    Destination interface name: si-2/2/0.10
    Test size: 20 probes
    Probe results:
      Response received
      Probe sent time: Fri Jan 11 11:48:50 2019
      Probe rcvd/timeout time: Fri Jan 11 11:48:50 2019

.....
Results over all tests:
      Probes sent: 218, Probes received: 218, Loss percentage: 0.000000
      Measurement: Round trip time
        Samples: 218, Minimum: 54 usec, Maximum: 59 usec, Average: 56 usec, Peak to peak: 5
usec, Stddev: 1 usec, Sum: 12160 usec

......
    Owner: c1, Test: t2
    server-address: 192.0.2.2, server-port: 862, Client address: 192.168.20.1, Client port: 58991
    TWAMP-Server-Status: Connected, Number-Of-Retries-With-TWAMP-Server: 12
    Reflector address: 192.0.2.2, Reflector port: 14962, Sender address: 192.168.20.1, sender-
port: 14962
    Routing Instance Name: IN
    Destination interface name: si-2/2/0.10
    Test size: 15 probes
    Probe results:
      Response received
      Probe sent time: Fri Jan 11 11:48:50 2019
      Probe rcvd/timeout time: Fri Jan 11 11:48:50 2019
      Rtt: 57 usec, Egress jitter: 2 usec, Ingress jitter: -3 usec,
......
Results over all tests:
      Probes sent: 168, Probes received: 168, Loss percentage: 0.000000
      Measurement: Round trip time
        Samples: 168, Minimum: 57 usec, Maximum: 59 usec, Average: 58 usec, Peak to peak: 2
usec, Stddev: 1 usec, Sum: 9691 usec
      Measurement: Positive egress jitter
        Samples: 124, Minimum: 0 usec, Maximum: 398 usec, Average: 11 usec, Peak to peak: 398
usec, Stddev: 61 usec, Sum: 1406 usec
      Measurement: Negative egress jitter
```

```
        Samples: 29, Minimum: 16 usec, Maximum: 431 usec, Average: 62 usec, Peak to peak: 415 
usec, Stddev: 74 usec, Sum: 1806 usec
      Measurement: Positive ingress jitter
        Samples: 30, Minimum: 0 usec, Maximum: 431 usec, Average: 60 usec, Peak to peak: 431 
usec, Stddev: 74 usec, Sum: 1811 usec
      Measurement: Negative ingress jitter
        Samples: 123, Minimum: 1 usec, Maximum: 397 usec, Average: 11 usec, Peak to peak: 396 
usec, Stddev: 61 usec, Sum: 1410 usec
      Measurement: Positive round trip jitter
        Samples: 125, Minimum: 0 usec, Maximum: 1 usec, Average: 0 usec, Peak to peak: 1 usec, 
Stddev: 0 usec, Sum: 42 usec
      Measurement: Negative round trip jitter
        Samples: 42, Minimum: 1 usec, Maximum: 1 usec, Average: 1 usec, Peak to peak: 0 usec, 
Stddev: 0 usec, Sum: 42 usec
```

RELATED DOCUMENTATION

tcp-keepcnt

tcp-keepintvl

tcp-keepidle

persistent-results

Understanding TWAMP Auto-Restart | **781**

CHAPTER 16

Managing License Server for Throughput Data Export

IN THIS CHAPTER

License Server Management for Throughput Data Export on MX Series Routers for NAT, Firewall, and Inline Flow Monitoring Services

IN THIS SECTION

To support our transition to software defined networking (SDN), Juniper Networks supports the Software Business Model Transformation, which includes new licensing, pricing, and branding strategies that make it easier for users to extract value from Juniper software solutions. This value of this approach is known as the Juniper Software Advantage (JSA), which provides the following benefits:

- Simple—Simple to buy, use, and manage rights
- Repeatable—License models which facilitates repeatable use among multiple hardware platforms and usage scenarios.
- Measurable—License fees based on easy to measure usage

Although the licensing of JSA products is trust-based, Juniper Networks might periodically audit the usage of its products. License Measurement Tool (LMT) is a technique that is used to compute the usage of individual Network Edge Products under JSA. MX Series routers need to define the mechanism for updating the LMT tool with information such as per-service throughput. For example, for services such as carrier-grade NAT and inline flow monitoring, the router needs to calculate per service throughput and update it in LMT.

On MX Series routers, the Routing Engine periodically sends query messages to every Service PIC on which the service, for which throughput collection is being performed, is configured to run. This polling is performed for all the services for which throughput measurement is enabled. Service PICs, upon receiving the query for a particular service, reply with the throughput measured during the last query interval, for that service. If a service PIC hosts multiple services, the Routing Engine sends separate throughput queries to that service PIC for all the services. If a service is configured on multiple services PICs, the Routing Engine aggregates the throughput values received from all of them and exports the aggregated throughput to the log collector in the predefined log format. The LMT application analyze these values from log collector, performs aggregation on values collected from all routers, and displays them in the LMT application.

You can configure the capability to transmit the throughput details per service for the Junos Address Aware (carrier-grade NAT) and Junos Traffic Vision (previously known as Jflow) in the last time interval to an external log collector. The default time interval at which the throughput data is sent is 300 seconds, which you can configure to suit your network needs. Multiple instances of the same service running on different PICs within a router are supported. If the same service is running on different PICs within a router, the router transmits the consolidated final throughput to the log collector or server. This functionality is supported on MX Series routers with MS-MCPs and MS-MICs, and also in the MX Series Virtual Chassis configuration. To configure the license server properties for throughput data to be transmitted for the defined services, such as NAT or stateful firewall, from the service PIC on the router to the external log collector, include the license-server statement at the [edit] hierarchy level. To specify the IP address of the license log server, include the `ip-address` *address* statement at the `[edit license-server]` hierarchy level. To configure the frequency of transmission of throughput date, include the `log-interval` *seconds* statement at the [edit license-server] hierarchy level. To specify the services for which throughput data collection must be performed, include the `services (jflow | cgnat | firewall)` statement at the `[edit license-server]` hierarchy level.

Throughput Measurement and Export

Throughput is defined as: “The network traffic throughput processed by juniper software in a second. It is represented as Mb/Sec (Megabits per second) or GB/sec (Gigabits per second). Throughput is measured as the 95th percentile of all the peaks measured in a quarter.” Service PICs keep track of the amount of data (in bits) processed by the various service plugins running on them. When a throughput query arrives from the Routing Engine, for a particular service, the Service PIC returns the value D/T mbps, in its reply, where:

- D is the amount of data (megabits) processed by that service since the previous query was received. If the query interval happens to be 300 seconds, for example, then D refers to the amount of data that was processed during the last 300 second interval. If the current query happens to be the very first query, for a particular service, then D represents the cumulative data bits processed so far, by that service.

- T is the time (seconds) that elapsed since the previous query was received. This is the query interval configured using the CLI interface. If the current query happens to be the very first query, for a particular service, then T represents the time that elapsed since that service started processing packets. For all subsequent queries, T equals the query interval.

The Routing Engine aggregates the throughput measured (in mbps) across all the Service PICs on which a particular Service is configured and exports it to the Log collector which performs the 95th percentile calculation.

RELATED DOCUMENTATION

License Enforcement

Verifying Junos OS License Installation (CLI)

show system license

Guidelines for Configuring an MX Series Router to Transmit Per-Service Throughput to an External Log Collector

Observe the following guidelines while configuring this functionality on MX Series routers with MS-MPCs and MS-MICs:

- If the syslog server is unreachable, the router cannot send information to the log collector.
- After a graceful Routing Engine switchover (GRES) procedure, the newly functioning active Routing Engine starts sending the data to the server after the configured time interval, which is similar to a reset operation. The time elapsed in the active interval and data before GRES are not preserved.
- The time range must be from 60 through 86400 seconds (24 hours).
- If the timer is not configured, the default value of 300 seconds is assumed.
- The throughput data can be sent only if a service is up and running.
- Only maximum throughput is transmitted for the last 300 seconds or the configured time interval.
- The throughput value must not be less than zero to enable transmission. The data is sent based on the timezone of the router.

- An acknowledgment mechanism for data sent to the log collector is not supported. The router does not receive any acknowledgement regarding whether the data is already written into the log collector.
- The router does not maintain throughput data beyond the configurable time interval.
- No mechanisms exist to track if the log collector is successfully receiving the sent data or if the log server is reachable.
- The time interval and log collector are common for all the services; you cannot configure a different period for collection of logs for each service or a different log collector for each service.
- You cannot clear the system throughput value using a CLI command. It is assumed that the throughput value is not cleared or changed from outside. Throughput must be calculated internally by services and must not be manually modified by a CLI.
- SNMP support for these values is not available.
- The log collector performs a 95 percentile calculation of throughput data. Syslogs are sent even in scaled system conditions to the log collector for the throughput data related to the configured services.
- The following is the format of the syslogs configured to be sent at the prescribed frequency:

```
<Date>    <Time> < time-zone> <Router_name> <Service_name> <Throughput_value> Throughput =
<Unit_Mbps/Gbps> in last <Time_Interval>
```

  An example is as follows:

```
Jan  8 08:49:57  America/Adak deuterium  CGNAT Throughput = 1500000 Mbps in last 300Sec
```

CHAPTER 17

Testing the Performance of Network Devices Using RFC 2544-Based Benchmarking

IN THIS CHAPTER

Understanding RFC 2544-Based Benchmarking Tests on MX Series Routers and SRX Series Firewalls

SUMMARY

The RFC 2544-based benchmarking test methodology assesses different parameters that are defined in service-level agreements (SLAs). By measuring the performance availability, transmission delay, link

bursts, and service integrity, a carrier provider can certify that the working parameters of the deployed Ethernet circuit comply with the SLA and other defined policies.

RFC 2544 defines a series of tests that can be used to describe the performance characteristics of a network-interconnecting device, such as a router, and outlines specific formats to report the results of the tests. These tests can be used to benchmark interconnected network devices and devise a guideline or a measurement pattern to analyze the health and efficiency of the network devices. These tests are the standard benchmarking tests for Ethernet networks and are known as RFC 2544-based benchmarking tests. These tests measure throughput, latency, frame loss rate, and bursty frames. The test methodology enables you to define various parameters such as the different frame sizes to be examined (64, 128, 256, 512, 1024, 1280, and 1518 bytes), the test time for each test iteration (10 seconds through 1,728,000 seconds), and the frame format (UDP-over-IP).

An RFC 2544-based benchmarking test is performed by transmitting test packets from a device that functions as the generator or the initiator (which is also called the originator). These packets are sent to a device that functions as a reflector, which receives and returns the packets to the initiator.

NOTE: MX Series routers and SRX Series Firewalls support only the reflector function in RFC 2544-based benchmarking tests. ACX Series routers support the initiator function in RFC 2544-based benchmarking tests. See "RFC 2544-Based Benchmarking Tests for ACX Routers Overview" on page 922.

NOTE: RFC 2544-based benchmarking tests support only UDP over IPv4 test traffic (unicast).

Starting from Junos OS Release 15.1, MX104 Series routers also perform verification of signatures on the received test frames. Starting in Junos OS Release 21.1R1, SRX300 and SRX550HM devices perform verification of signatures on the received test frames. By default, when the router or device receives a test packet that does not have the signature pattern, the packet is dropped. If you generate test traffic using a third-party vendor tool instead of an ACX Series router, you can disable signature verification. To disable signature verification, configure the `disable-signature-check` statement at the [edit services rpm rfc2544-benchmarking tests test-name test-name] hierarchy level.

For MX80 and MX104 Universal Routing Platforms and, starting from Junos OS Release 16.1, MX240, MX480, and MX960 Universal Routing Platforms with MPC1 (MX-MPC1-3D), MPC2 (MX-MPC2-3D), and the 16-port 10-Gigabit Ethernet MPC (MPC-3D-16XGE-SFP), support the reflector function and the corresponding benchmarking tests.

Starting from Junos OS Release 17.1R1, the reflector function and the corresponding benchmarking tests are supported on MX Series routers with MPC3E (MX-MPC3E-3D), MPC3E-NG (MX-MPC3E-3D-

NG), MPC4E (MPC4E-3D-32XGE-SFPP and MPC4E-3D-2CGE-8XGE), MPC5E (MPC5E-40G10G, MPC5EQ-40G10G, MPC5E-100G10G, and MPC5EQ-100G10G), and MPC6E (MX2K-MPC6E).

Starting in Junos OS Release 20.2R1, the RFC 2544-based benchmarking tests are supported on MX240, MX480, and MX960 routers with the MPC10E (MPC10E-15C-MRATE and MPC10E-10C-MRATE) line card and on MX2010 and MX2020 routers with the MX2K-MPC11E line card.

Junos OS Release 20.3R1 extends support for the RFC 2544-based benchmarking tests onto the MX240, MX480, and MX960 routers with the MPC7E-MRATE or MPC7E-10G line card, onto the MX2008, MX2010, and MX2020 routers with the MX2K-MPC8E or MX2K-MPC9E line card, and onto the MX204 and MX10003 (with the LC2103 card) routers.

Starting in Junos OS Release 21.1R1, the IPv4 Layer 3 reflector function and the corresponding benchmarking tests are supported on the SRX300 and SRX550HM devices.

NOTE: To configure RFC2544-based benchmarking tests on MX Series routers, see "Enabling Support for RFC 2544-Based Benchmarking Tests on MX Series Routers" on page 814.

Table 133 on page 797 describes the different MX network topologies in which the benchmarking test is supported.

Table 133: Supported MX Network Topologies for RFC2544 Benchmarking Tests

Service Type	Traffic Direction	Mode	Initial Release on MX104 Series Routers	Initial Release on MX204, MX2008, and MX10003 Series Routers	Initial Release on MX240, MX480, and MX960 Series Routers	Initial Release on MX2010 and MX2020 Series Routers	Whether the Benchmarking Test Is Supported
E-Line (family `bridge`)	(UNI) Egress (UNI) Ingress	Port Port, VLAN	14.2R1 (E-Line family `bridge`) 17.1R1	20.3R1 20.3R1	16.1R1 17.1R1	20.2R1 20.2R1	Supported

Table 133: Supported MX Network Topologies for RFC2544 Benchmarking Tests *(Continued)*

Service Type	Traffic Direction	Mode	Initial Release on MX104 Series Routers	Initial Release on MX204, MX2008, and MX10003 Series Routers	Initial Release on MX240, MX480, and MX960 Series Routers	Initial Release on MX2010 and MX2020 Series Routers	Whether the Benchmarking Test Is Supported
E-LAN (family `bridge` and family `vpls`)	(UNI) Egress (UNI) Ingress	Port Port, VLAN	14.2R1 (E-LAN family `bridge`) 15.1R1 (E-LAN family `vpls`) 17.1R1	20.3R1 20.3R1	16.1R1 17.1R1	20.2R1 20.2R1	Supported
E-Line (family `ccc`)	Ingress Egress	Port Port, VLAN	13.3R1 (E-Line pseudowire)	20.3R1 20.3R1	16.1R1	20.2R1	Supported
IP Services (family `inet`)	NNI	Port Port, VLAN	13.3R1	20.3R1	16.1R1	20.2R1	Supported

NOTE: You can configure a total of four simultaneous active reflection sessions. The four active reflection sessions can be of the same type or can be a combination of the different types of reflection sessions. For instance, you can configure either four IPv4 reflection sessions or one session each for pseudowire reflection, VPLS reflection, Layer 2 reflection, and IPv4 reflection. The maximum reflection bandwidth supported is 4 Gbps in a standalone test condition. Starting in Junos OS Release 20.2R1, MPC10E and MX2K-MPC11E support a maximum reflection bandwidth of 100 Gbps.

Table 134 on page 799 lists the interfaces and the reflection type on which the benchmarking tests are supported.

Table 134: Supported Interfaces for RFC2544 Benchmarking Tests

Type of Reflection	Gigabit Interfaces (ge)	Aggregated Interfaces (ae)	40G/100G interfaces (et) for MPC10E and MX2K-MPC11E	10G Interfaces (xe)	Pseudo Interfaces (irb, lt, vt, lo0, and others)
IPv4	Yes	Yes	Yes	Yes	No
Pseudowire ingress	Yes	Yes	Yes	yes	No
Pseudowire egress	Yes	Yes (starting in Junos OS Release 15.1)	Yes	Yes (starting in Junos OS Release 15.1)	No
Layer 2 bridge	Yes	Yes	Yes	Yes	No
Layer 2 VPLS	Yes	Yes	Yes	Yes	No

All active RFC2544-based benchmarking tests are stopped when any of the following events take place:

- System events, such as Packet Forwarding Engine restarts, Routing Engine restarts, and so on.
- Test interface change events, such as deactivation and reactivation of the interface, disabling and enabling of the interface, child link events for aggregated interfaces and so on.

  After the benchmarking tests are stopped, the test states of the tests are removed and the user can restart the same test. Other ongoing tests on other interfaces are not interrupted.

NOTE: RFC2544-based benchmarking tests are not supported during an unified in-service software upgrade (ISSU) or a graceful Routing Engine switchover (GRES).

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
21.1R1	Starting from Junos OS Release 21.1R1, the IPv4 Layer 3 reflector function and the corresponding benchmarking tests are supported on the SRX300 and SRX550HM devices.

20.3R1	Junos OS Release 20.3R1 extends support for the RFC 2544-based benchmarking tests onto the MX240, MX480, and MX960 routers with the MPC7E-MRATE or MPC7E-10G line card, onto the MX2008, MX2010, and MX2020 routers with the MX2K-MPC8E or MX2K-MPC9E line card, and onto the MX204 and MX10003 (with the LC2103 card) routers.
20.2R1	Starting in Junos OS Release 20.2R1, the RFC 2544-based benchmarking tests are supported on MX240, MX480, and MX960 routers with the MPC10E (MPC10E-15C-MRATE and MPC10E-10C-MRATE) line card and on MX2010 and MX2020 routers with the MX2K-MPC11E line card.
17.1R1	Starting from Junos OS Release 17.1R1, the reflector function and the corresponding benchmarking tests are supported on MX Series routers with MPC3E (MX-MPC3E-3D), MPC3E-NG (MX-MPC3E-3D-NG), MPC4E (MPC4E-3D-32XGE-SFPP and MPC4E-3D-2CGE-8XGE), MPC5E (MPC5E-40G10G, MPC5EQ-40G10G, MPC5E-100G10G, and MPC5EQ-100G10G), and MPC6E (MX2K-MPC6E).
16.1	For MX80 and MX104 Universal Routing Platforms and, starting from Junos OS Release 16.1, MX240, MX480, and MX960 Universal Routing Platforms with MPC1 (MX-MPC1-3D), MPC2 (MX-MPC2-3D), and the 16-port 10-Gigabit Ethernet MPC (MPC-3D-16XGE-SFP), support the reflector function and the corresponding benchmarking tests.
15.1	Starting from Junos OS Release 15.1, MX104 Series routers also perform verification of signatures on the received test frames.

RELATED DOCUMENTATION

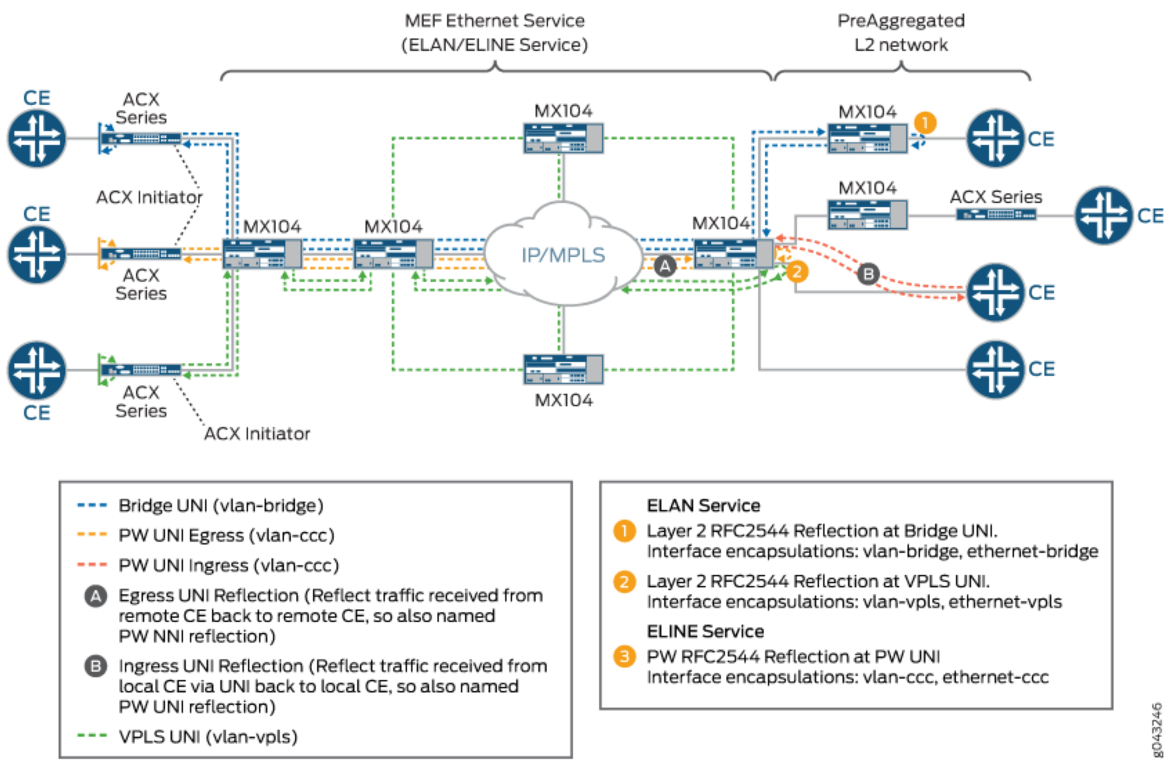
Configuring an RFC 2544-Based Benchmarking Test | 806
Supported RFC 2544-Based Benchmarking Statements on MX Series Routers | 804
Enabling Support for RFC 2544-Based Benchmarking Tests on MX Series Routers | 814

Understanding RFC2544-Based Benchmarking Tests for E-LAN and E-Line Services on MX Series Routers

NOTE: MX Series routers support only the reflector function in RFC2544-based benchmarking tests.

The Metro Ethernet Forum (MEF) defines two Ethernet service types—E-LAN and E-Line—and specifies the associated service attributes and parameters. These services can be supported within the Metro Ethernet Network (MEN) and also supported over different transport technologies such as SONET, MPLS, and so on. Juniper Networks ACX Series routers, MX80, MX104 Series routers and MX240, MX480, and MX960 Series routers with MPC1, MPC2, and 16-port 10-Gigabit Ethernet MPC provide support for Layer 2 E-LAN and E-Line services reflection. Figure 67 on page 801 shows a sample topology for the E-LAN and E-Line reflection supported on MX104 Series routers.

Figure 67: E-LAN And E-Line Reflection in a metro Solution



Starting in Junos OS Release 15.1, MX104 Series routers support RFC2544-based benchmarking tests for Layer 2 reflection (E-Line service) by using pseudowires (Layer 2 circuit and L2VPN). Starting in Junos OS Release 16.1, MX80 Series routers and MX240, MX480, and MX960 Series routers with MPC1, MPC2, and 16-port 10-gigabit Ethernet MPC support RFC2544-based benchmarking tests for Layer 2 reflection (E-Line service). E-Line provides transparent data transport. You can configure RFC2544-based benchmarking tests for both ingress and egress direction on the customer edge (CE) facing interface of family type `ccc` for an Ethernet pseudowire.

NOTE: To configure RFC2544-based benchmarking tests on MX240, MX480, MX960 Series routers with MPC1, MPC2, and the 16-port 10-Gigabit Ethernet MPC, see "Enabling Support for RFC 2544-Based Benchmarking Tests on MX Series Routers" on page 814.

Starting in Junos OS Release 15.1, MX104 routers support RFC2544-based benchmarking tests for Layer 2 reflection (E-LAN service) by using VPLS and basic bridge domains. In Junos OS Release 14.2 and earlier, only basic bridge domains are used. Starting in Junos OS Release 16.1, MX80 Series routers and MX240, MX480, and MX960 Series routers with MPC1, MPC2, and 16-port 10-gigabit Ethernet MPC support RFC2544-based benchmarking tests for Layer 2 reflection (E-LAN service) by using VPLS and basic bridge domains. VPLS enables geographically dispersed sites to share an Ethernet broadcast domain by connecting sites across an MPLS network. All sites appear to be in the same Ethernet LAN though traffic travels across the MPLS network. Both LDP-based VPLS and BGP-based VPLS are supported. RFC2544-based benchmarking and performance measurement testing for Layer 2 E-LAN services (`bridge`/ `VPLS`) is supported on unicast traffic in egress direction only.

During the benchmarking tests, the initiator or generator transmits a test packet (unicast) to a reflector. The reflector receives and reflects the test packet back to the initiator. The test packet is an UDP-over-IP packet with a source and destination MAC address.

In a E-LAN service, the Layer 2 traffic reflection session is identified by the source MAC address, the destination MAC address, and the egress interface (logical interface). By default, RFC2544-based benchmarking tests are performed when there is no other service traffic. This mode of operation is known as out-of-service mode. The default service mode for the reflecting egress interface for an E-LAN service is also out-of-service mode. In out-of-service mode, while the test is running, all the data traffic (other than test traffic) sent to and from the test interface under test is interrupted. If the test is activated on a logical interface, all the traffic sent to and from the logical interface is interrupted. However, if there are other logical interfaces on the UNI port, the traffic sent to and from those logical interfaces is not interrupted. Control protocol peering is not interrupted whereas pass through control protocol packets such as end-to-end CFM sessions are interrupted. If you do not want the control protocol packets interrupted, you can configure the E-LAN service mode as in-service mode. In the in-service mode, while the test is running, the rest of the data traffic flow sent to and from the UNI port under test on the service is not interrupted. Both peering and pass through control protocols are not interrupted.

In an E-Line service, the reflection session is identified by the egress interface which is the logical interface. On activation of reflection on a logical interface, the traffic received on the logical interface is reflected. You can specify the type of traffic you want reflected by specifying the EtherType (specifies the protocol transported). If you do not specify the EtherType, all traffic is reflected. System does not explicitly block other traffic on the test interface during E-line service. You can block non-test traffic using firewall filters.

By default, for E-LAN services, the reflector swaps MAC addresses. The reflector swaps the source and destination MAC addresses and sends the packet back to the initiator. By default, for E-Line services, the reflector does not swap MAC addresses. Table 135 on page 803 describes the MAC address swapping behavior for the service types.

Table 135: MAC Address Swapping Behavior for E-LAN and E-Line Services

Family	Direction	Default Behavior	User-configurable
`bridge`	Egress	MAC address swap (E-LAN service type) No MAC address swap (E-Line service type)	No Yes
`vpls`	Egress	MAC address swap (E-LAN service type)	No
`ccc`	Egress	No MAC address swap	Yes (starting in Junos OS Release 15.1)
	Ingress	MAC address swap	No

By default, the IP addresses and UDP ports are not modified. Optionally, you can configure the reflector to swap the source and destination IP address and the source and destination UDP ports.

You can configure an ACX Series router to operate as an initiator as well as a reflector. The MX104 Series router can be configured to operate only as a reflector.

Starting in Junos OS Release 15.1, MX104 Series routers support the specification of the protocol transported in the Ethernet frame. Starting in Junos OS Release 16.1, MX80 Series routers and MX240, MX480, and MX960 Series routers with MPC1, MPC2, and 16-port 10-gigabit Ethernet MPC also support the specification of the protocol transported in the Ethernet frame. To specify the EtherType (specifies the protocol transported) used for reflection of the test frames, use the `reflect-etype` command. If you do not specify the EtherType, all EtherTypes are reflected.

NOTE: The maximum reflection bandwidth supported is 4 Gbps. Because RFC2544 reflection shares system bandwidth with other loopback services such as tunnel services, you must manage the sharing of bandwidth for performing RFC2544-based performance tests.

NOTE: RFC2544-based benchmarking tests are not supported during unified in-service software upgrade (ISSU) and graceful Routing Engine switchover (GRES).

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
16.1	Starting in Junos OS Release 15.1, MX104 Series routers support the specification of the protocol transported in the Ethernet frame.
15.1	Starting in Junos OS Release 15.1, MX104 Series routers support RFC2544-based benchmarking tests for Layer 2 reflection (E-Line service) by using pseudowires (Layer 2 circuit and L2VPN).
15.1	Starting in Junos OS Release 15.1, MX104 routers support RFC2544-based benchmarking tests for Layer 2 reflection (E-LAN service) by using VPLS and basic bridge domains.

RELATED DOCUMENTATION

Understanding RFC 2544-Based Benchmarking Tests on MX Series Routers and SRX Series Firewalls | **795**

Supported RFC 2544-Based Benchmarking Statements on MX Series Routers | **804**

Example: Configuring RFC2544-Based Benchmarking Tests on an MX104 Router for Layer 2 E-LAN Services in Bridge Domains | **855**

disable-signature-check

reflect-etype

Supported RFC 2544-Based Benchmarking Statements on MX Series Routers

Table 136 on page 805 lists the reflector-specific configuration statements that are supported on the MX Series routers. Note that en dash (–) specified in the Initial Release on MX Series routers column denotes that the command is not supported.

Table 136: Supported RFC2544-Based Benchmarking Reflector Statements on MX Series

Statement	Options	Initial Release on MX104 Series Routers	Intial Release on MX240, MX480, MX960 Series Routers
destination-ipv4-address	–	13.3R1	16.1R1
destination-mac-address	–	14.2R1	16.1R1
destination-udp-port	–	13.3R1	16.1R1
direction	(egress | ingress)	13.3R1	16.1R1
disable-signature-check	–	15.1R1	16.1R1
family	(ccc | inet) (bridge | ccc | inet) (vpls)	13.3R1 14.2R1 15.1R1	16.1R1
in-service	–	14.2R1	16.1R1
ip-swap	–	14.2R1	16.1R1
mode	reflect	13.3R1	16.1R1
reflect-etype	–	15.1R1	16.1R1
reflect-mode	(mac-swap | no-mac-swap)	14.2R1	16.1R1
service-type	(eline | elan)	14.2R1	16.1R1
source-ipv4-address	–	13.3R1	16.1R1

Table 136: Supported RFC2544-Based Benchmarking Reflector Statements on MX Series *(Continued)*

Statement	Options	Initial Release on MX104 Series Routers	Intial Release on MX240, MX480, MX960 Series Routers
`source-mac-address`	–	14.2R1	16.1R1
`source-udp-port`	–	13.3R1	16.1R1
`test-interface`	–	13.3R1	16.1R1
`udp-tcp-port-swap`	–	14.2R1	16.1R1

RELATED DOCUMENTATION

Understanding RFC 2544-Based Benchmarking Tests on MX Series Routers and SRX Series Firewalls | 795

Configuring an RFC 2544-Based Benchmarking Test | 806

Example: Configuring RFC2544-Based Benchmarking Tests on an MX104 Router for Layer 2 E-LAN Services in Bridge Domains | 855

Configuring an RFC 2544-Based Benchmarking Test

IN THIS SECTION

You can configure a benchmarking test to detect and measure performance attributes, such as throughput, latency, frame loss, and bursty or back-to-back frames, of network devices. An RFC 2544-based benchmarking test is performed by transmitting test packets from a device that functions as the initiator and terminator of the test. These packets are sent to a device that functions as the reflector, which receives and returns the packets back to the initiator.

NOTE: The test configuration is applied only when you start the test. If you update the test configuration during the test, you have to start the test again for the updated configuration to take effect.

The following devices support RFC 2544-based benchmarking tests in either the initiator/terminator or reflector role, according to which families they support:

Table 137: RFC 2544-Based Benchmarking Tests by Role and Family Supported

Platform	family			
	inet	bridge	ccc	vpls
Initiator and Terminator Role				
ACX Series (except for ACX5000 and ACX7000)	x	x	x	
Reflector Role				
ACX Series (except for ACX5000 and ACX7000)	x	x	x	
ACX5000 Series		x	x	
ACX7000 Series	x			
MX Series	x	x	x	x
SRX300 Series and SRX550HM	x			

The family type for the test is configured with the `family` *`name`* statement at the `[edit services rpm rfc2544-benchmarking tests test-name` *`name`*`]` hierarchy level.

You must configure a test profile and reference the test profile in a unique test name that defines the parameters for the test to be performed on a certain device. However, the test profile is required when the test mode is configured as initiation and termination. The `test-profile` parameter is disregarded when the test mode is configured as reflection. MX Series routers and SRX Series Firewalls support only the reflection function in the RFC 2544-based benchmarking tests. A reflection service does not use the parameters specified in the test profile.

> **NOTE**: To configure RFC2544-based benchmarking tests on MX240, MX480, MX960 Series routers with MPC1, MPC2, and the 16-port 10-Gigabit Ethernet MPC, see "Enabling Support for RFC 2544-Based Benchmarking Tests on MX Series Routers" on page 814.

This chapter describes how to configure a test name for an RFC 2544-based benchmarking test on an MX Series router for Layer 3 IPv4, Ethernet pseudowire, and Layer 2 bridge networks. For SRX Series Firewalls, you can only configure Layer 3 IPv4 reflection (`family inet` only).

Configuring a Test Name for an RFC 2544-Based Benchmarking Test for a IPv4 Network

You can configure a test name by including the `test-name` *`test-name`* statement at the `[edit services rpm rfc2544-benchmarking]` hierarchy level. In the test name, you can configure attributes of the test iteration, such as the address family (type of service, IPv4 or Ethernet), the logical interface, and test duration that are used for a benchmarking test to be run.

To configure a test name and define its attributes for an IPv4 network:

1. In configuration mode, go to the `[edit services]` hierarchy level.

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

2. Configure a instance.

   ```
   [edit services]
   user@host# edit rpm
   ```

3. Configure an RFC 2544-based benchmarking test for the RPM instance.

   ```
   [edit services rpm]
   user@host# edit rfc2544-benchmarking
   ```

4. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

5. Specify the test mode for the packets that are sent during the benchmarking test. The `reflect` option causes the test frames to be reflected on the IPv4 network.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set mode reflect
```

6. Configure the address type family for the benchmarking test. The `inet` option indicates that the test is run on an IPv4 service.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set family inet
```

7. Configure the destination IPv4 address for the test packets. This parameter is required only if you configure IPv4 family `inet`. If you do not configure the destination IPv4 address, the default value of 192.168.1.20 is used.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set destination-ipv4-address address
```

8. Specify the UDP port of the destination to be used in the UDP header for the generated frames. If you do not specify the UDP port, the default value of 4041 is used.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set destination-udp-port port-number
```

9. (Optional) Specify the source IPv4 address to be used in generated test frames. If you do not configure the source IPv4 address for `inet` family, the source address of the interface is used to transmit the test frames.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set source-ipv4-address address
```

10. Specify the UDP port of the source to be used in the UDP header for the generated frames. If you do not specify the UDP port, the default value of 4041 is used.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set source-udp-port port-number
```

11. Specify the logical interface on which the RFC 2544-based benchmarking test is run. If you configure an `inet` family and the test mode to reflect the frames back on the sender from the other end, then the logical interface is used as the interface to enable the reflection service (reflection is performed on the packets entering the specified interface). If you not configure the logical interface for reflection test mode, then a lookup is performed on the source IPv4 address to determine the interface that hosts the address.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-interface interface-name
```

Configuring a Test Name for an RFC 2544-Based Benchmarking Test for an Ethernet Pseudowire

You can configure a test name by including the `test-name` *`test-name`* statement at the `[edit services rpm rfc2544-benchmarking]` hierarchy level. In the test name, you can configure attributes of the test iteration, such as the address family (type of service, IPv4 or Ethernet), the logical interface, and test duration, that are used for a benchmarking test to be run. The test name combined with the test profile represent a single real-time performance monitoring (RPM) configuration instance.

To configure a test name and define its attributes for an Ethernet Pseudowire:

1. In configuration mode, go to the `[edit services]` hierarchy level.

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

2. Configure an RPM service instance.

```
[edit services]
user@host# edit rpm
```

3. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@host# edit rfc2544-benchmarking
```

4. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

5. Specify the test mode for the packets that are sent during the benchmarking test. The `reflect` option causes the test frames to be reflected on the Ethernet pseudowire.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set mode reflect
```

6. Configure the address type family for the benchmarking test. The `ccc` option indicates that the test is run on a CCC or Ethernet pseudowire service.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set family ccc
```

7. Specify the direction of the interface on which the test must be run. This parameter is valid only for a family. To enable the test to be run in the egress direction of the interface (network-to-network interface (NNI)), use the `egress` option. To enable the test to be run in the ingress direction of the interface (user-to-network interface (UNI)), use the `ingress` option.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set direction (egress | ingress)
```

8. (Optional) Specify the source IPv4 address to be used in generated test frames. If you do not configure the source IPv4 address for family, the default value of 192.168.1.10 is used.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set source-ipv4-address address
```

9. Specify the logical interface on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-interface interface-name
```

Configuring a Test Name for an RFC 2544-Based Benchmarking Test for a Layer 2 E-LAN Service in Bridge Domain

You can configure a test name by including the `test-name` *`test-name`* statement at the `[edit services rpm rfc2544-benchmarking]` hierarchy level. In the test name, you can configure attributes of the test iteration, such as the address family (bridge), the logical interface, and test duration, that are used for a benchmarking test to be run. The test name combined with the test profile represent a single real-time performance monitoring (RPM) configuration instance.

To configure a test name and define its attributes for a layer 2 E-LAN service in Bridge domains:

1. In configuration mode, go to the `[edit services]` hierarchy level.

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

2. Configure an RPM service instance.

   ```
   [edit services]
   user@host# edit rpm
   ```

3. Configure an RFC 2544-based benchmarking test for the RPM instance.

   ```
   [edit services rpm]
   user@host# edit rfc2544-benchmarking
   ```

4. Define a name for the test—for example, l2b-test1. The test name identifier can be up to 32 characters in length.

   ```
   [edit services rpm rfc2544-benchmarking]
   user@host# edit tests test-name l2b-test1
   ```

5. Specify the source and destination MAC addresses of the test packet. Both these parameters are valid only for the bridge family.

```
[edit services rpm rfc2544-benchamrking tests test-name l2b-test1]
user@host# set source-mac-address address destination-mac-address address
```

6. Specify the service type under test. This parameter is applicable only for the bridge family.

```
[edit services rpm rfc2544-benchamrking tests test-name l2b-test1]
user@host# set service-type elan
```

7. Specify the test mode for the packets that are sent during the benchmarking test. The `reflect` option causes the test frames to be reflected over the Layer 2 bridge.

```
[edit services rpm rfc2544-benchmarking tests test-name l2b-test1]
user@host# set mode reflect
```

8. Configure the address type family for the benchmarking test. The `bridge` option indicates that the test is run on a E-LAN service over a bridge domain.

```
[edit services rpm rfc2544-benchmarking tests test-name l2b-test1]
user@host# set family bridge
```

9. Specify the direction of the interface on which the test must be run. This parameter is valid only for a family. To enable the test to be run in the egress direction of the interface (network-to-network interface (NNI)), use the `egress` option. To enable the test to be run in the ingress direction of the interface (user-to-network interface (UNI)), use the `ingress` option.

```
[edit services rpm rfc2544-benchmarking tests test-name l2b-test1]
user@host# set direction egress
```

10. Specify the logical interface on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name l2b-test1]
user@host# set test-interface interface-name
```

RELATED DOCUMENTATION

Understanding RFC 2544-Based Benchmarking Tests on MX Series Routers and SRX Series Firewalls | 795

Example: Configuring an RFC 2544-Based Benchmarking Test on an MX104 Router for UNI Direction of Ethernet Pseudowires | 830

Example: Configuring an RFC 2544-Based Benchmarking Test on an MX104 Router for NNI Direction of Ethernet Pseudowires | 842

Example: Configuring RFC2544-Based Benchmarking Tests on an MX104 Router for Layer 2 E-LAN Services in Bridge Domains | 855

Example: Configure an RFC 2544-Based Benchmarking Test on an MX104 Router for Layer 3 IPv4 Services | 816

Enabling Support for RFC 2544-Based Benchmarking Tests on MX Series Routers

> **NOTE**: MX Series routers support only the reflector function in RFC 2544-based benchmarking tests.

RFC 2544 defines a series of tests that can be used to describe the performance characteristics of a network-interconnecting device, such as a router, and outlines specific formats to report the results of the tests. These tests can be used to benchmark interconnected network devices and devise a guideline or a measurement pattern to analyze the health and efficiency of the network devices. These tests are known as RFC 2544-based benchmarking tests and are supported on MX80, MX104, MX240, MX480, MX960, and MX2010 routers with MPC1 (MX-MPC1-3D), MPC2 (MX-MPC2-3D), and the 16-port 10-Gigabit Ethernet MPC (MPC-3D-16XGE-SFP). Starting from Junos OS Release 17.1R1, the RFC 2544-based benchmarking tests are supported on MX Series routers with MPC3E (MX-MPC3E-3D), MPC3E-NG (MX-MPC3E-3D-NG), MPC4E (MPC4E-3D-32XGE-SFPP and MPC4E-3D-2CGE-8XGE), MPC5E (MPC5E-40G10G, MPC5EQ-40G10G, MPC5E-100G10G, and MPC5EQ-100G10G), and MPC6E (MX2K-MPC6E).Starting from Junos OS Release 20.2R1, the RFC 2544-based benchmarking tests are supported on MX240, MX480, and MX960 routers with MPC10E (MPC10E-15C-MRATE and MPC10E-10C-MRATE) line card and MX2010 and MX2020 routers with MX2K-MPC11E line card.

Junos OS Release 20.3R1 extends support for the RFC 2544-based benchmarking tests onto the MX240, MX480, and MX960 routers with the MPC7E-MRATE or MPC7E-10G line card, onto the MX2008, MX2010, and MX2020 routers with the MX2K-MPC8E or MX2K-MPC9E line card, and onto the MX204 and MX10003 (with the LC2103 card) routers.

NOTE: On MX104 and MX80 Series routers that have a single fixed FPC, this configuration is not required.

To enable support for RFC 2544-based benchmarking tests on MX Series routers:

1. In configuration mode, go to the [`edit chassis fpc` *`fpc-slot-number`*] hierarchy level.

```
[edit]
user@host# edit chassis fpc fpc-slot-number
```

2. Enable support for service-level agreement (SLA) monitoring services and RFC-based benchmarking tests:

```
[edit chassis fpc fpc-slot-number]
user@host# set slamon-services rfc2544
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.3R1	Junos OS Release 20.3R1 extends support for the RFC 2544-based benchmarking tests onto the MX240, MX480, and MX960 routers with the MPC7E-MRATE or MPC7E-10G line card, onto the MX2008, MX2010, and MX2020 routers with the MX2K-MPC8E or MX2K-MPC9E line card, and onto the MX204 and MX10003 (with the LC2103 card) routers.
20.2R1	Starting from Junos OS Release 20.2R1, the RFC 2544-based benchmarking tests are supported on MX240, MX480, and MX960 routers with MPC10E (MPC10E-15C-MRATE and MPC10E-10C-MRATE) line card and MX2010 and MX2020 routers with MX2K-MPC11E line card.
17.1R1	Starting from Junos OS Release 17.1R1, the RFC 2544-based benchmarking tests are supported on MX Series routers with MPC3E (MX-MPC3E-3D), MPC3E-NG (MX-MPC3E-3D-NG), MPC4E (MPC4E-3D-32XGE-SFPP and MPC4E-3D-2CGE-8XGE), MPC5E (MPC5E-40G10G, MPC5EQ-40G10G, MPC5E-100G10G, and MPC5EQ-100G10G), and MPC6E (MX2K-MPC6E).

RELATED DOCUMENTATION

Understanding RFC 2544-Based Benchmarking Tests on MX Series Routers and SRX Series Firewalls | 795

Configuring an RFC 2544-Based Benchmarking Test | 806

slamon-services

Example: Configure an RFC 2544-Based Benchmarking Test on an MX104 Router for Layer 3 IPv4 Services

IN THIS SECTION

Requirements

NOTE: MX Series routers support only the reflector function in RFC2544-based benchmarking tests. This example uses the MX104 3D Universal Edge Router as the reflector. You can also configure benchmarking tests on MX80 routers and on MX240, MX480, and MX960 routers with MPC1, MPC2, and 16-port 10-Gigabit Ethernet MPC from Junos OS Release 16.1 or later. To configure RFC2544-based benchmarking tests on MX240, MX480, MX960 routers, see "Enabling Support for RFC 2544-Based Benchmarking Tests on MX Series Routers" on page 814.

NOTE: This example is not applicable for ACX7100, ACX5448, ACX5048, and ACX5096 routers because they can only be configured as reflectors, not initiators.

This example uses the following hardware and software components:

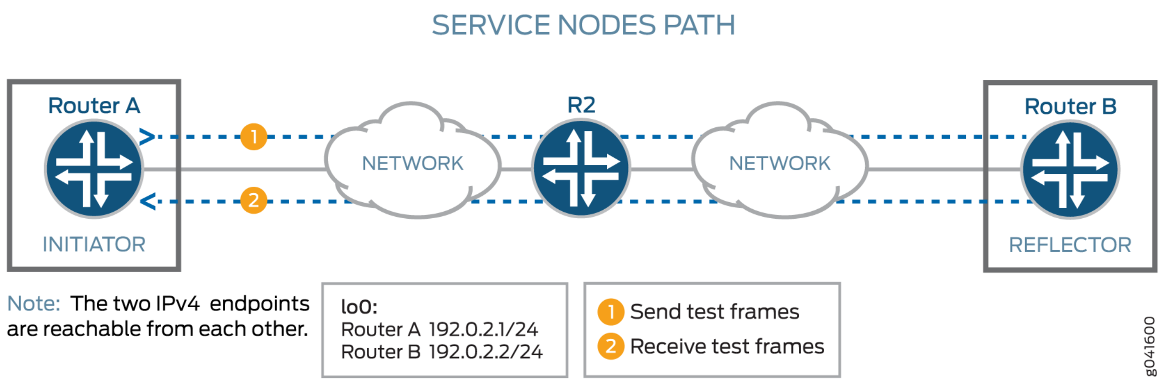
- An MX104 router (reflector)
- An ACX Series router (initiator)
- Junos OS Release 13.3 or later

Overview

Consider a sample topology in which a router, Router A (ACX), functions as an initiator and terminator of the test frames for an RFC 2544-based benchmarking test. Router A is connected over a Layer 3 network to another router, Router B (MX104), which functions as a reflector to reflect back the test frames it receives from Router A. IPv4 is used for transmission of test frames over the Layer 3 network. This benchmarking test is used to compute the IPv4 service parameters between Router A and Router B. Logical interfaces on both the routers are configured with IPv4 addresses to measure the performance attributes, such as throughput, latency, frame loss, and bursty frames, of network devices for the IPv4 service.

Figure 68 on page 817 shows the sample topology to perform an RFC 2544 test for a Layer 3 IPv4 Service.

Figure 68: RFC 2544-Based Benchmarking Test for a Layer 3 IPv4 Service



Configuration

IN THIS SECTION

In this example, you configure the benchmarking test for a Layer 3 IPv4 service that is between interface ge-0/0/0 on Router A and interface ge-0/0/4 on Router B to detect and analyze the performance of the interconnecting routers. You do not configure a test profile on Router B, because it operates as a reflector. You must configure the reflector (Router B) before you configure the initiator (Router A), because the reflector needs to be already configured and the tests running before you start tests on the initiator. If you start the tests on the initiator first, then all the packets sent are lost until you start the tests on the reflector.

CLI Quick Configuration

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

Configure Benchmarking Test Parameters on Router B

```
set interfaces ge-0/0/4 unit 0 family inet address 192.0.2.2/24
set services rpm rfc2544-benchmarking tests test-name test1 test-interface ge-0/0/4.0
set services rpm rfc2544-benchmarking tests test-name test1 mode reflect
set services rpm rfc2544-benchmarking tests test-name test1 family inet
set services rpm rfc2544-benchmarking tests test-name test1 destination-ipv4-address 192.0.2.2
set services rpm rfc2544-benchmarking tests test-name test1 destination-udp-port 4001
set services rpm rfc2544-benchmarking tests test-name test1 source-ipv4-address 192.0.2.1
```

Configure Benchmarking Test Parameters on Router A

```
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.1/24
set services rpm rfc2544-benchmarking profiles test-profile throughput test-type throughput
set services rpm rfc2544-benchmarking profiles test-profile throughput packet-size 64
set services rpm rfc2544-benchmarking profiles test-profile throughput bandwidth-kbps 1000
set services rpm rfc2544-benchmarking tests test-name test1 test-profile throughput
set services rpm rfc2544-benchmarking tests test-name test1 test-interface ge-0/0/0.0
set services rpm rfc2544-benchmarking tests test-name test1 mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name test1 family inet
set services rpm rfc2544-benchmarking tests test-name test1 destination-ipv4-address 192.0.2.2
set services rpm rfc2544-benchmarking tests test-name test1 destination-udp-port 4001
set services rpm rfc2544-benchmarking tests test-name test1 source-ipv4-address 192.0.2.1
```

Configure Benchmarking Test Parameters on Router B

Step-by-Step Procedure

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

To configure the test parameters on Router B:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

```
[edit]
user@RouterB# edit interfaces
```

2. Configure the interface on which the test must be run.

```
[edit interfaces]
user@RouterB# edit ge-0/0/4
```

3. Configure a logical unit and specify the protocol family as `inet`.

```
[edit interfaces ge-0/0/4]
 user@RouterB# edit unit 0 family inet
```

4. Specify the address for the logical interface.

```
[edit interfaces ge-0/0/4 unit 0 family inet]
user@RouterB# set address 192.0.2.2/24
```

5. Go to the top level of the configuration mode.

```
[edit interfaces ge-0/0/4 unit 0]
user@RouterB# top
```

6. In configuration mode, go to the `[edit services rpm rfc2544-benchmarking]` hierarchy level.

```
[edit]
user@RouterB# edit services rpm rfc2544-benchmarking
```

7. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@RouterB# edit tests test-name test1
```

8. Specify the logical interface, ge-0/0/4.0, on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# set test-interface ge-0/0/4.0
```

9. Specify `reflect` as the test mode for the packets that are sent during the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# set mode reflect
```

10. Configure the address type family, `inet`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# set family inet
```

11. Configure the destination IPv4 address for the test packets as 192.0.2.2. The destination IPv4 address configured on the reflector must match the destination IPv4 address configured on the initiator. If you configure 192.0.2.1 instead, you get this error message: `error: test test1 - Could not determine local interface for address 192.0.2.1`.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set destination-ipv4-address 192.0.2.2
```

12. Specify the UDP port of the destination to be used in the UDP header for the generated frames as 4001.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set destination-udp-port 4001
```

13. Configure the source IPv4 address for the test packets.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set source-ipv4-address 192.0.2.1
```

14. Go to the top level of the configuration mode.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# top
```

15. Commit the configuration.

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

16. Confirm the configuration. If the output does not contain the configuration below, repeat the configuration instructions in this example to correct it.

```
[edit interfaces]
  ge-0/0/4 {
      unit 0 {
          family inet {
              address 192.0.2.2/24;
          }
       }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      # Note, When in reflector mode, test profile is not needed
      tests {
          test-name test1 {
```

```
                test-interface ge-0/0/4.0;
                mode reflect;
                family inet;
                destination-ipv4-address 192.0.2.2;
                destination-udp-port 4001;
                source-ipv4-address 192.0.2.1
            }
        }
    }
```

17. Exit to operational mode.

```
[edit]
user@RouterB# exit
user@RouterB>
```

18. Start the benchmarking test on the reflector.

```
user@host> test services rpm rfc2544-benchmarking test test1 start
```

After the test is successfully completed at the initiator, you can stop the test at the reflector by entering the `test services rpm rfc2544-benchmarking test test1 stop` command.

Configure Benchmarking Test Parameters on Router A

Step-by-Step Procedure

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

To configure the test parameters on Router A:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

```
[edit]
user@RouterA# edit interfaces
```

2. Configure the interface on which the test must be run.

```
[edit interfaces]
user@RouterA# edit ge-0/0/0
```

3. Configure a logical unit and specify the protocol family.

```
[edit interfaces ge-0/0/0]
 user@RouterA# edit unit 0 family inet
```

4. Specify the address for the logical interface.

```
[edit interfaces ge-0/0/0 unit 0 family inet]
user@RouterA# set address 192.0.2.1/24
```

5. Go to the top level of the configuration mode.

```
[edit interfaces ge-0/0/0 unit 0]
user@RouterA# top
```

6. In configuration mode, go to the `[edit services rpm rfc2544-benchmarking]` hierarchy level.

```
[edit]
user@RouterA# edit services rpm rfc2544-benchmarking
```

7. Define a name for a test profile—for example, throughput.

```
[edit services rpm rfc2544-benchmarking]
user@RouterA# edit profiles test-profile throughput
```

8. Configure the type of test to be performed as throughput.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@RouterA# set test-type throughput
```

9. Specify the size of the test packet as 64 bytes.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@RouterA# set packet-size 64
```

10. Define the theoretical maximum bandwidth for the test in kilobits per second, with a value from 1,000 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@RouterA# set bandwidth-kbps 1000
```

11. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@RouterA# up
```

12. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles]
user@RouterA# up
```

13. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@RouterA# edit tests test-name test1
```

14. Specify the name of the test profile—for example, throughput—to be associated with a particular test name.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set test-profile throughput
```

15. Specify the logical interface, ge-0/0/0.0, on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set test-interface ge-0/0/0.0
```

16. Specify the test mode for the packets that are sent during the benchmarking test as initiate and terminate.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set mode initiate-and-terminate
```

17. Configure the address type family, `inet`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set family inet
```

18. Configure the destination IPv4 address for the test packets.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set destination-ipv4-address 192.0.2.2
```

19. Specify the UDP port of the destination to be used in the UDP header for the generated frames as 4001.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set destination-udp-port 4001
```

20. Configure the source IPv4 address for the test packets.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set source-ipv4-address 192.0.2.1
```

21. Go to the top level of the configuration mode.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# top
```

22. Commit the configuration.

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

23. Confirm the configuration. If the output does not contain the configuration below, repeat the configuration instructions in this example to correct it.

```
[edit]
user@RouterA# show
[edit interfaces]
  ge-0/0/0 {
      unit 0 {
          family inet {
              address 192.0.2.1/24;
          }
       }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      profiles {
          test-profile throughput {
              test-type throughput
              packet-size 64;
              bandwidth-kbps 1000;
          }
      }

      tests {
          test-name test1 {
              test-profile throughput;
              interface ge-0/0/0.0;
              mode initiate-and-terminate;
              family inet;
```

```
                destination-ipv4-address 192.0.2.2
                destination-udp-port 4001;
                source-ipv4-address 192.0.2.1
            }
        }
    }
```

24. Exit to operational mode.

```
[edit]
user@RouterA# exit
user@RouterA>
```

25. Start the benchmarking test on the initiator.

```
user@RouterA> test services rpm rfc2544-benchmarking test test1 start
```

After the test successfully completes, it automatically stops at the initiator. Once the test is successfully completed at the initiator, you can stop the test at the reflector by entering the `test services rpm rfc2544-benchmarking test test1 stop` command on Router B in operational mode.

Results

If you have not done so already, confirm your configuration on Router A and Router B by entering the `show` command in configuration mode at the [edit interfaces] and [edit services rpm] hierarchy levels. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

Configuration for Benchmarking Test Parameters on Router A:

```
[edit interfaces]
  ge-0/0/0 {
      unit 0 {
          family inet {
              address 192.0.2.1/24;
          }
      }
  }

  [edit services rpm]
```

```
  rfc2544-benchmarking {
      profiles {
          test-profile throughput {
              test-type throughput
              packet-size 64;
              bandwidth-kbps 1000;
          }
      }

      tests {
          test-name test1 {
              test-profile throughput;
              test-interface ge-0/0/0.0;
              mode initiate-and-terminate;
              family inet;
              destination-ipv4-address 192.0.2.2
              destination-udp-port 4001;
              source-ipv4-address 192.0.2.1
          }
      }
  }
```

Configuration for Benchmarking Test Parameters on Router B:

```
[edit interfaces]
  ge-0/0/4 {
      unit 0 {
          family inet {
              address 192.0.2.2/24;
          }
      }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      # Note, When in reflector mode, test profile is not needed
      tests {
          test-name test1 {
              test-interface ge-0/0/4.0;
              mode reflect;
              family inet;
```

```
            destination-ipv4-address 192.0.2.2;
            destination-udp-port 4001;
            source-ipv4-address 192.0.2.1
        }
    }
}
```

Verify the Results of the Benchmarking Test for Layer 3 IPv4 Services

IN THIS SECTION

Examine the results of the benchmarking test that is performed on the configured service between Router A and Router B.

Verify the Benchmarking Test Results

Purpose

Verify that the necessary and desired statistical values are displayed for the benchmarking test that is run on the configured service between Router A and Router B.

Action

In operational mode, enter the `show services rpm rfc2544-benchmarking (aborted-tests | active-tests | completed-tests | summary)` command, on either the initiator or the reflector, to display information about the results of each category or state of the RFC 2544-based benchmarking test, such as terminated tests, active tests, and completed tests, for each real-time performance monitoring (RPM) instance.

RELATED DOCUMENTATION

Understanding RFC 2544-Based Benchmarking Tests on MX Series Routers and SRX Series Firewalls | 795

Configuring an RFC 2544-Based Benchmarking Test | 806

Example: Configuring an RFC 2544-Based Benchmarking Test on an MX104 Router for UNI Direction of Ethernet Pseudowires

SUMMARY

This example shows how to configure the RFC2544-based benchmarking test for the user-to-network interface (UNI) direction of an Ethernet pseudowire service that is enabled between two routers. The test detects and analyzes the performance of the interconnected routers.

IN THIS SECTION

Requirements

MX Series routers support only the reflector function in RFC2544-based benchmarking tests. This example uses the MX104 3D Universal Edge Router as the reflector. You can also configure benchmarking tests on MX80 Series routers and MX240, MX480, and MX960 Series routers with MPC1, MPC2, and 16-port 10-gigabit Ethernet MPC from Junos OS Release 16.1 or later. To configure RFC2544-based benchmarking tests on MX240, MX480, MX960 Series routers, see "Enabling Support for RFC 2544-Based Benchmarking Tests on MX Series Routers" on page 814.

This example uses the following hardware and software components:

- An MX104 (reflector)
- An ACX Series router (initiator)
- Junos OS Release 13.3 or later

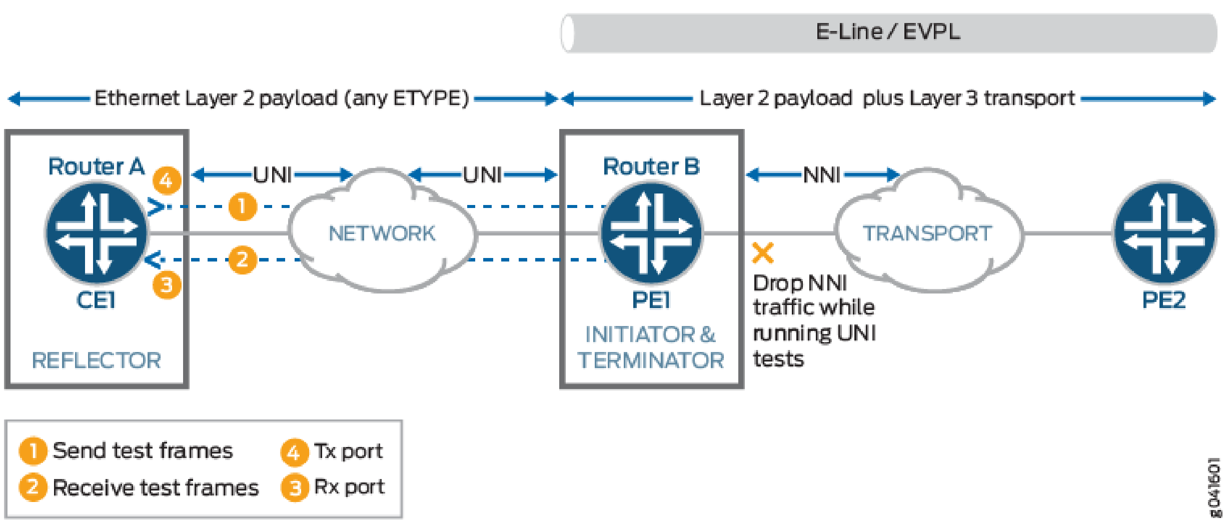
Overview

Consider a sample topology in which a router, Router A (MX104), functions as a reflector of the test frames for an RFC 2544-based benchmarking test. The logical customer edge (CE)-facing interface and `inet` family are configured on Router A. Router A is not part of a pseudowire and therefore, a Layer 3 family configuration is required on it. Router A, which is a customer edge device CE1 is connected to Router B (ACX), which functions as a provider edge device PE1 over an Ethernet pseudowire in the UNI direction with EtherType or Layer 2 Ethernet payload. The logical interface, family, and UNI direction are configured on Router B. Router B or PE1 is connected over an Ethernet pseudowire in the NNI direction to a provider edge device at the remote site, PE2. The link between CE1 and PE1 is an Ethernet Layer 2 network and it can be configured with any EtherType value. The link between PE1 and PE2 is an Ethernet line (E-Line) or an Ethernet Private Line (EPL) that has Layer 2 payload and Layer 3 transport sent over it. Router B or PE1 functions as an initiator and terminator of the test frames that are sent to Router A and reflected back from it.

This benchmarking test is used to compute the performance attributes in the user-to-network interface (UNI) direction of an Ethernet pseudowire service between Router A and Router B. Data traffic arriving from a network-to-network interface (NNI) toward the customer edge is ignored while the test is in progress. Packets from the CE are not sent toward the NNI because all packets are assumed to be test probes.

Figure 69 on page 831 shows the sample topology to perform an RFC 2544 test for the UNI direction of an Ethernet pseudowire service.

Figure 69: RFC 2544-Based Benchmarking Test for UNI Direction of an Ethernet Pseudowire



Configuration

IN THIS SECTION

In this example, you configure the benchmarking test for the UNI direction of an Ethernet pseudowire service that is enabled between two routers to detect and analyze the performance of the interconnected routers.

CLI Quick Configuration

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

Configuring Benchmarking Test Parameters on Router B (Initiator and Terminator)

```
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 vlan-id 101
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.1/24
set services rpm rfc2544-benchmarking profiles test-profile throughput test-type throughput
set services rpm rfc2544-benchmarking profiles test-profile throughput packet-size 64
set services rpm rfc2544-benchmarking profiles test-profile throughput test-duration 20m
set services rpm rfc2544-benchmarking profiles test-profile throughput bandwidth-kbps 500
set services rpm rfc2544-benchmarking tests test-name test1 interface ge-0/0/0.0
set services rpm rfc2544-benchmarking tests test-name test1 test-profile throughput
set services rpm rfc2544-benchmarking tests test-name test1 mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name test1 family inet
set services rpm rfc2544-benchmarking tests test-name test1 dest-address 192.0.2.2
set services rpm rfc2544-benchmarking tests test-name test1 udp-port 4001
```

Configuring Benchmarking Test Parameters on Router A (Reflector)

```
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 0 encapsulation vlan-ccc
set interfaces ge-0/0/4 unit 0 vlan-id 101
set services rpm rfc2544-benchmarking tests test-name test1 interface ge-0/0/4.0
set services rpm rfc2544-benchmarking tests test-name test1 mode reflect
set services rpm rfc2544-benchmarking tests test-name test1 mode family ccc
set services rpm rfc2544-benchmarking tests test-name test1 direction uni
```

Configuring Benchmarking Test Parameters on Router B (Initiator and Terminator)

Step-by-Step Procedure

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

To configure the test parameters on Router B:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

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

2. Configure the interface on which the test must be run.

   ```
   [edit interfaces]
   user@host# edit ge-0/0/0
   ```

3. Configure VLAN tagging for transmission and reception of 802.1Q VLAN-tagged frames.

   ```
   [edit interfaces ge-0/0/0]
    user@host# set vlan-tagging
   ```

4. Configure a logical unit and specify the protocol family as `inet`.

   ```
   [edit interfaces ge-0/0/0]
    user@host# edit unit 0 family inet
   ```

5. Specify the address for the logical interface.

   ```
   [edit interfaces ge-0/0/0 unit 0 family inet]
   user@host# set address 192.0.2.1/24
   ```

6. Configure the VLAN ID on the logical interface as 101.

```
[edit interfaces ge-0/0/0 unit 0]
user@host# set vlan-id 101
```

7. Go to the top level of the configuration command mode.

```
[edit interfaces ge-0/0/0 unit 0]
user@host# top
```

8. In configuration mode, go to the `[edit services]` hierarchy level.

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

9. Configure a real-time performance monitoring service (RPM) instance.

```
[edit services]
user@host# edit rpm
```

10. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@host# edit rfc2544-benchmarking
```

11. Define a name for a test profile—for example, throughput.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit profiles test-profile throughput
```

12. Configure the type of test to be performed as throughput.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type throughput
```

13. Specify the size of the test packet as 64 bytes.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type packet-size 64
```

14. Specify the period for which the test is to be performed in hours, minutes, or seconds by specifying a number followed by the letter h (for hours), m (for minutes), or s (for seconds). In this example, you configure the period as 20 minutes.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type test-duration 20m
```

15. Define the theoretical maximum bandwidth for the test in kilobits per second, with a value from 1 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type bandwidth-kbps 500
```

16. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# up
```

17. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles]
user@host# up
```

18. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

19. Specify the name of the test profile—for example, throughput—to be associated with a particular test name.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-profile throughput
```

20. Specify the logical interface, ge-0/0/0.0, on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-interface ge-0/0/0.0
```

21. Specify the test mode for the packets that are sent during the benchmarking test as `initiate-and-terminate`.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set mode initiate-and-terminate
```

22. Configure the address type family, `inet`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set family inet
```

23. Configure the destination IPv4 address for the test packets as 192.0.2.2.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set dest-address 192.0.2.2
```

24. Specify the UDP port of the destination to be used in the UDP header for the generated frames as 4001.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set udp-port 4001
```

Verify the Configuration and Commit

In configuration mode, confirm your configuration on Router B by entering the show command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

Benchmarking Test Parameters on Router B:

```
[edit]
user@host# show
[edit interfaces]
  ge-0/0/0 {
      vlan-tagging;
      unit 0 {
          vlan-id 101;
          family inet {
              address 192.0.2.1/24;
          }
      }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      profiles {
          test-profile throughput {
              test-type throughput
              packet-size 64;
              test-duration 20m;
              bandwidth-kbps 500;
          }
      }

      tests {
          test-name test1 {
              interface ge-0/0/0.0;
              test-profile throughput;
              mode initiate-and-terminate;
              family inet;
              dest-address 192.0.2.2
              udp-port 4001;
          }
```

```
    }
}
```

When the configuration is correct, commit the configuration:

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

Configuring Benchmarking Test Parameters on Router A (Reflector)

Step-by-Step Procedure

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

To configure the test parameters on Router A:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

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

2. Configure the interface on which the test must be run.

   ```
   [edit interfaces]
   user@host# edit ge-0/0/4
   ```

3. Configure VLAN tagging for transmission and reception of 802.1Q VLAN-tagged frames.

   ```
   [edit interfaces ge-0/0/4]
    user@host# set vlan-tagging
   ```

4. Configure a logical unit for the interface.

   ```
   [edit interfaces ge-0/0/4]
    user@host# edit unit 0
   ```

5. Specify the encapsulation for Ethernet VLAN circuits.

```
[edit interfaces ge-0/0/4 unit 0]
 user@host# set encapsulation vlan-ccc
```

6. Configure the VLAN ID as 101 on the logical interface.

```
[edit interfaces ge-0/0/4 unit 0]
user@host# set vlan-id 101
```

7. Go to the top level of the configuration command mode.

```
[edit interfaces ge-0/0/4 unit 0]
user@host# top
```

8. In configuration mode, go to the `[edit services]` hierarchy level.

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

9. Configure a real-time performance monitoring service (RPM) instance.

```
[edit services]
user@host# edit rpm
```

10. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@host# edit rfc2544-benchmarking
```

11. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

12. Specify the logical interface on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-interface ge-0/0/4.0
```

13. Specify `reflect` as the test mode for the packets that are sent during the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set mode reflect
```

14. Configure the address type family, `ccc`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set family ccc
```

15. Specify the direction of the interface on which the test must be run, which is UNI in this example.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set direction uni
```

Verify the Configuration and Commit

In configuration mode, confirm your configuration on Router A by entering the `show` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

Benchmarking Test Parameters on Router A:

```
[edit]
user@host# show
[edit interfaces]
  ge-0/0/4 {
      vlan-tagging;
      unit 0 {
          encapsulation vlan-ccc;
          vlan-id 101;
      }
  }
```

```
[edit services rpm]
rfc2544-benchmarking {
    # Note, When in reflector mode, test profile is not needed
    tests {
        test-name test1 {
            interface ge-0/0/4.0;
            mode reflect;
            family ccc;
            direction uni;
        }
    }
}
```

When the configuration is correct, commit the configuration.

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

Run the Benchmarking Test

- Start the benchmarking test on the reflector (Router A).

  ```
  user@host> test services rpm rfc2544-benchmarking test test1 start
  ```

  The test stops automatically at the initiator (Router B) when the test successfully completes. After the test successfully completes at the initiator, stop the test at the reflector (Router A).

  ```
  user@host> test services rpm rfc2544-benchmarking test test1 stop
  ```

Verifying the Benchmarking Test Results

IN THIS SECTION

Examine the results of the benchmarking test that is performed on the configured service between Router A and Router B.

Purpose

Verify that the necessary and desired statistical values are displayed for the benchmarking test that is run on the configured service between Router A and Router B.

Action

In operational mode, enter the `show services rpm rfc2544-benchmarking (aborted-tests | active-tests | completed-tests | summary)` command to display information about the results of each category or state of the RFC 2544-based benchmarking test, such as terminated tests, active tests, and completed tests, for each real-time performance monitoring (RPM) instance.

Meaning

The output displays the details of the benchmarking test that was performed. For more information about the `show services rpm rfc2544-benchmarking` operational command, see `show services rpm rfc2544-benchmarking` in the CLI Explorer.

RELATED DOCUMENTATION

Understanding RFC 2544-Based Benchmarking Tests on MX Series Routers and SRX Series Firewalls | 795

Configuring an RFC 2544-Based Benchmarking Test | 806

Example: Configuring an RFC 2544-Based Benchmarking Test on an MX104 Router for NNI Direction of Ethernet Pseudowires

IN THIS SECTION

This example shows how to configure the benchmarking test for a network-to-network interface (NNI) direction of an Ethernet pseudowire service.

Requirements

> **NOTE**: MX Series routers support only the reflector function in RFC2544-based benchmarking tests. This example uses the MX104 3D Universal Edge Router as the reflector. You can also configure benchmarking tests on MX80 Series routers and MX240, MX480, and MX960 Series routers with MPC1, MPC2, and 16-port 10-gigabit Ethernet MPC from Junos OS Release 16.1 or later. To configure RFC2544-based benchmarking tests on MX240, MX480, MX960 Series routers, see "Enabling Support for RFC 2544-Based Benchmarking Tests on MX Series Routers" on page 814.

This example uses the following hardware and software components:

- An MX104 (reflector)
- An ACX Series router (initiator)
- Junos OS Release 13.3 or later

Overview

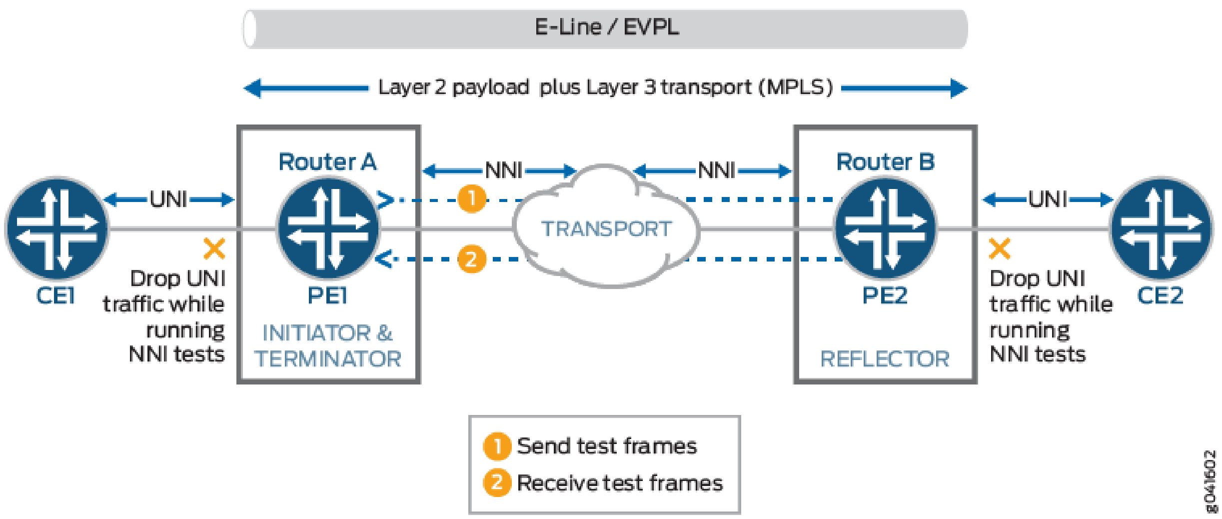
Consider a sample topology in which a router, Router A (ACX), functions as an initiator and terminator of the test frames for an RFC 2544-based benchmarking test. Router A operates as a provider edge devicePE1, which is connected to a customer edge device CE1 on one side and over an Ethernet pseudowire to another router Router B (MX104), which functions as a reflector to reflect back the test frames it receives from Router A. Router B operates as a provider edge device, PE2, which is the remote router located at the other side of the service provider core. The UNI direction of CE1 is connected to the NNI direction of PE1. An MPLS tunnel connects PE1 and PE2 over the Ethernet pseudowire or the Ethernet line (E-Line).

> **NOTE**: When pseudowire reflection is enabled on an interface, the router does not block the ingress or egress traffic through the test interface. To block other data traffic, you must explicitly configure firewall filters.

This benchmarking test is used to compute the performance attributes in the network-to-network interface (NNI) direction of an Ethernet pseudowire service between Router A and Router B. The logical interface under test on Router A is the CE1 interface with UNI as the direction, and the logical interface under test on Router B is the CE2 interface with NNI as the direction. Data traffic arriving from UNI toward NNI is ignored while the test is in progress. Packets from NNI are not sent toward the customer edge because all packets are assumed to be test frames. The family and NNI direction are configured on routers A and B.

Figure 70 on page 844 shows the sample topology to perform an RFC 2544 test for the NNI direction of an Ethernet pseudowire service.

Figure 70: RFC 2544-Based Benchmarking Test for NNI Direction of an Ethernet Pseudowire



Configuration

IN THIS SECTION

In this example, you configure the benchmarking test for the NNI direction of an Ethernet pseudowire service that is enabled between two routers to detect and analyze the performance of the interconnecting routers.

CLI Quick Configuration

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

Configuring Benchmarking Test Parameters on Router A

```
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 encapsulation vlan-ccc
set interfaces ge-0/0/0 unit 0 vlan-id 101
set services rpm rfc2544-benchmarking profiles test-profile throughput test-type throughput
set services rpm rfc2544-benchmarking profiles test-profile throughput packet-size 64
set services rpm rfc2544-benchmarking profiles test-profile throughput test-duration 20
set services rpm rfc2544-benchmarking profiles test-profile throughput bandwidth-kbps 500
set services rpm rfc2544-benchmarking tests test-name test1 interface ge-0/0/0.1
set services rpm rfc2544-benchmarking tests test-name test1 test-profile throughput
set services rpm rfc2544-benchmarking tests test-name test1 mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name test1 family ccc
set services rpm rfc2544-benchmarking tests test-name test1 direction egress
```

Configuring Benchmarking Test Parameters on Router B

```
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 0 encapsulation vlan-ccc
set interfaces ge-0/0/4 unit 0 vlan-id 101
set services rpm rfc2544-benchmarking tests test-name test1 interface ge-0/0/4.1
set services rpm rfc2544-benchmarking tests test-name test1 mode reflect
set services rpm rfc2544-benchmarking tests test-name test1 mode family ccc
set services rpm rfc2544-benchmarking tests test-name test1 direction ingress
```

Configuring Benchmarking Test Parameters on Router

Step-by-Step Procedure

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

To configure the test parameters on Router A:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

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

2. Configure the interface on which the test must be run.

   ```
   [edit interfaces]
   user@host# edit ge-0/0/0
   ```

3. Configure VLAN tagging for transmission and reception of 802.1Q VLAN-tagged frames.

   ```
   [edit interfaces ge-0/0/0]
    user@host# set vlan-tagging
   ```

4. Configure a logical unit for the interface.

   ```
   [edit interfaces ge-0/0/0]
    user@host# edit unit 0
   ```

5. Specify the encapsulation for Ethernet VLAN circuits.

   ```
   [edit interfaces ge-0/0/0 unit 0]
    user@host# set encapsulation vlan-ccc
   ```

6. Configure the VLAN ID on the logical interface.

```
[edit interfaces ge-0/0/0 unit 0]
user@host# set vlan-id 101
```

7. Go to the top level of the configuration command mode.

```
[edit interfaces ge-0/0/0 unit 0]
user@host# top
```

8. In configuration mode, go to the `[edit services]` hierarchy level.

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

9. Configure a real-time performance monitoring service (RPM) instance.

```
[edit services]
user@host# edit rpm
```

10. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@host# edit rfc2544-benchmarking
```

11. Define a name for a test profile—for example, throughput.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit profiles test-profile throughput
```

12. Configure the type of test to be performed as throughput.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type throughput
```

13. Specify the size of the test packet as 64 bytes.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type packet-size 64
```

14. Specify the period—for example, 20 minutes—for which the test is to be performed in hours, minutes, or seconds by specifying a number followed by the letter h (for hours), m (for minutes), or s (for seconds).

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type test-duration 20m
```

15. Define the theoretical maximum bandwidth for the test in kilobits per second, with a value from 1 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type bandwidth-kbps 500
```

16. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# up
```

17. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles]
user@host# up
```

18. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

19. Specify the name of the test profile—for example, throughput—to be associated with a particular test name.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-profile throughput
```

20. Specify the logical interface, ge-0/0/0.1, on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-interface ge-0/0/0.1
```

21. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set mode initiate-and-terminate
```

22. Configure the address type family, `ccc`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set family ccc
```

23. Specify the direction of the interface on which the test must be run, which is egress in this example.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set direction egress
```

Configuring Benchmarking Test Parameters on Router B

Step-by-Step Procedure

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

To configure the test parameters on Router B:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

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

2. Configure the interface on which the test must be run.

```
[edit interfaces]
user@host# edit ge-0/0/4
```

3. Configure VLAN tagging for transmission and reception of 802.1Q VLAN-tagged frames.

```
[edit interfaces ge-0/0/4]
 user@host# set vlan-tagging
```

4. Configure a logical unit for the interface.

```
[edit interfaces ge-0/0/4]
 user@host# edit unit 0
```

5. Specify the encapsulation for Ethernet VLAN circuits.

```
[edit interfaces ge-0/0/4 unit 0]
 user@host# set encapsulation vlan-ccc
```

6. Configure the VLAN ID on the logical interface.

```
[edit interfaces ge-0/0/4 unit 0]
user@host# set vlan-id 101
```

7. Go to the top level of the configuration command mode.

```
[edit interfaces ge-0/0/4 unit 0]
user@host# top
```

8. In configuration mode, go to the `[edit services]` hierarchy level.

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

9. Configure a real-time performance monitoring service (RPM) instance.

```
[edit services]
user@host# edit rpm
```

10. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@host# edit rfc2544-benchmarking
```

11. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

12. Specify the logical interface, ge-0/0/4.1, on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-interface ge-0/0/4.1
```

NOTE: When pseudowire reflection is enabled on an interface, the router does not block the ingress or egress traffic through the test interface. To block other data traffic, you must explicitly configure firewall filters.

13. Specify `reflect` as the test mode for the packets that are sent during the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set mode reflect
```

14. Configure the address type family, `ccc`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set family ccc
```

15. Specify the direction of the interface on which the test must be run, which is ingress in this example.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set direction ingress
```

16. Start the benchmarking test on the reflector.

```
user@host> test services rpm rfc2544-benchmarking test test1 start
```

After the test is successfully completed at the initiator, you can stop the test at the reflector by entering the `test services rpm rfc2544-benchmarking test test1 stop` command.

Results

In configuration mode, confirm your configuration on Router A and Router B by entering the `show` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

Benchmarking Test Parameters on Router A:

```
[edit interfaces]
  ge-0/0/0 {
      vlan-tagging;
      unit 0 {
          encapsulation vlan-ccc;
          vlan-id 101;
      }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      profiles {
          test-profile throughput {
              test-type throughput
```

```
            packet-size 64;
            test-duration 20m;
            bandwidth-kbps 500;
        }
    }

    tests {
        test-name test1 {
            interface ge-0/0/0.1;
            test-profile throughput;
            mode initiate-and-terminate;
            family ccc;
            direction egress;
        }
    }
}
```

Benchmarking Test Parameters on Router B:

```
[edit interfaces]
  ge-0/0/4 {
      vlan-tagging;
      unit 0 {
          encapsulation vlan-ccc;
          vlan-id 101;
      }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      # Note, When in reflector mode, test profile is not needed
      tests {
          test-name test1 {
              interface ge-0/0/4.1;
              mode reflect;
              family ccc;
              direction egress;
          }
```

```
        }
    }
```

After you have configured the device, enter the `commit` command in configuration mode.

Verifying the Results of the Benchmarking Test for NNI Direction of an Ethernet Pseudowire Service

IN THIS SECTION

Examine the results of the benchmarking test that is performed on the configured service between Router A and Router B.

Verifying the Benchmarking Test Results

Purpose

Verify that the necessary and desired statistical values are displayed for the benchmarking test that is run on the configured service between Router A and Router B.

Action

In operational mode, enter the `show services rpm rfc2544-benchmarking (aborted-tests | active-tests | completed-tests | summary)` command to display information about the results of each category or state of the RFC 2544-based benchmarking test, such as terminated tests, active tests, and completed tests, for each real-time performance monitoring (RPM) instance.

Meaning

The output displays the details of the benchmarking test that was performed. For more information about the `show services rpm rfc2544-benchmarking` operational command, see `show services rpm rfc2544-benchmarking` in the CLI Explorer.

RELATED DOCUMENTATION

Understanding RFC 2544-Based Benchmarking Tests on MX Series Routers and SRX Series Firewalls | 795

Configuring an RFC 2544-Based Benchmarking Test | 806

Example: Configuring RFC2544-Based Benchmarking Tests on an MX104 Router for Layer 2 E-LAN Services in Bridge Domains

IN THIS SECTION

This example shows how to configure benchmarking tests for the Layer 2 E-LAN services in bridge domains. The example covers the four basic tests: throughput, frame-loss, back-to-back, and latency.

Requirements

NOTE: MX Series routers support only the reflector function in RFC2544-based benchmarking tests. This example uses the MX104 3D Universal Edge Router as the reflector. You can also configure benchmarking tests on MX80 Series routers and MX240, MX480, and MX960 Series routers with MPC1, MPC2, and 16-port 10-gigabit Ethernet MPC from Junos OS Release 16.1 or later. To configure RFC2544-based benchmarking tests on MX240, MX480, MX960 Series routers, see "Enabling Support for RFC 2544-Based Benchmarking Tests on MX Series Routers" on page 814.

This example uses the following hardware and software components:

- An MX104 (reflector)
- An ACX Series router (initiator)
- Junos OS Release 14.2 or later for MX Series routers

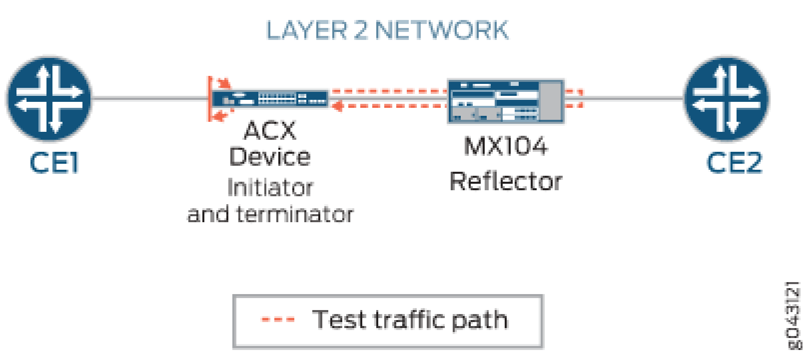
Overview

Consider a sample topology in which an ACX Series router functions as an initiator and terminator of the test frames for an RFC2544-based benchmarking test. ACX Series router is connected to a customer edge device CE1, on one side and is connected over a Layer 2 network to an MX104 Series router. The MX104 Series router functions as a reflector to reflect the test frames it receives from the ACX Series initiator back to the initiator. The MX04 Series router is also connected to a customer edge device CE2.

> **NOTE**: When Layer 2 reflection is enabled on an interface, filters are configured internally to block the ingress and egress traffic except test traffic through the test interface.

Figure 71 on page 856 shows the sample topology to perform all four RFC2544-based benchmarking tests (throughput, back-to-back frames, latency, and frame-loss) for the UNI direction on a Layer 2 bridge network.

Figure 71: Layer 2 Reflection Simple Topology



On the ACX Series router, ge-1/2/1.0 is the Layer 2 NNI interface and ge-1/1/3.0 is the Layer 2 UNI interface. On the MX104 Series router, ge-1/1/6.0 is the Layer 2 NNI interface and ge-1/1/5.0 is the Layer 2 UNI interface. The benchmarking tests are used to compute the performance attributes for an E-LAN service on a bridge domain.

> **NOTE**: Test packets can be identified using the destination MAC address, source MAC address, and test interface. Both tagged and untagged interfaces are supported. For tagged interfaces, the test interface is the VLAN sub interface. For untagged interfaces, the physical port represents the test interface. Traffic through other VLAN sub

interfaces, present in the same physical port, is not affected when you configure the benchmarking test on one of the sub interfaces.

Configuration

IN THIS SECTION

In this example, you configure the benchmarking tests for the UNI direction for an E-LAN service on a Layer 2 bridge domain that is enabled between two routers to detect and analyze the performance of the interconnected routers. In this example, we start by configuring the ACX Series router. On the ACX Series router, you first configure each test by specifying the test profile, the test attributes, and then define the test by associating the test with the test profile with the relevant attributes. You can then configure the interface. On the MX104 Series router, you perform the same steps. However, a few attributes such as the outer VLAN ID, source UDP port, destination UDP port, the duration of each iteration, and their values are only applicable to the initiator or the ACX Series router.

NOTE: When you configure the Layer 2 reflection, you can specify the service type under test as ELINE if you want to simulate an ELINE service using bridge encapsulation.

CLI Quick Configuration

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

Configuring Benchmarking Test Parameters on the ACX Series Router

```
set services rpm rfc2544-benchmarking profiles test-profile tput test-type throughput
set services rpm rfc2544-benchmarking profiles test-profile tput packet-size 128
set services rpm rfc2544-benchmarking profiles test-profile tput bandwidth-kbps 900000
set services rpm rfc2544-benchmarking profiles test-profile b2bt test-type back-back-frames
set services rpm rfc2544-benchmarking profiles test-profile b2bt packet-size 512
set services rpm rfc2544-benchmarking profiles test-profile b2bt bandwidth-kbps 950000
set services rpm rfc2544-benchmarking profiles test-profile lty test-type latency
set services rpm rfc2544-benchmarking profiles test-profile lty packet-size 512
set services rpm rfc2544-benchmarking profiles test-profile lty bandwidth-kbps 1000000
set services rpm rfc2544-benchmarking profiles test-profile frloss test-type frame-loss
set services rpm rfc2544-benchmarking profiles test-profile frloss packet-size 1600
set services rpm rfc2544-benchmarking profiles test-profile frloss bandwidth-kbps 1000000
set services rpm rfc2544-benchmarking tests test-name tput-test test-profile tput
set services rpm rfc2544-benchmarking tests test-name tput-test source-mac-address
00:00:5e:00:53:11
set services rpm rfc2544-benchmarking tests test-name tput-test destination-mac-address
00:00:5e:00:53:22
set services rpm rfc2544-benchmarking tests test-name tput-test ovlan-id 400
set services rpm rfc2544-benchmarking tests test-name tput-test service-type elan
set services rpm rfc2544-benchmarking tests test-name tput-test mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name tput-test family bridge
set services rpm rfc2544-benchmarking tests test-name tput-test direction egress
set services rpm rfc2544-benchmarking tests test-name tput-test source-udp-port 200
set services rpm rfc2544-benchmarking tests test-name tput-test destination-udp-port 200
set services rpm rfc2544-benchmarking tests test-name tput-test test-iterator-duration 20
set services rpm rfc2544-benchmarking tests test-name tput-test test-interface ge-1/1/3.0
set services rpm rfc2544-benchmarking tests test-name b2b-test test-profile b2bt
set services rpm rfc2544-benchmarking tests test-name b2b-test source-mac-address
00:00:5e:00:53:11
set services rpm rfc2544-benchmarking tests test-name b2b-test destination-mac-address
00:00:5e:00:53:22
set services rpm rfc2544-benchmarking tests test-name b2b-test ovlan-id 400
set services rpm rfc2544-benchmarking tests test-name b2b-test service-type elan
set services rpm rfc2544-benchmarking tests test-name b2b-test mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name b2b-test family bridge
set services rpm rfc2544-benchmarking tests test-name b2b-test direction egress
set services rpm rfc2544-benchmarking tests test-name b2b-test test-iterator-duration 20
set services rpm rfc2544-benchmarking tests test-name b2b-test test-interface ge-1/1/3.0
set services rpm rfc2544-benchmarking tests test-name lty-test test-profile lty
set services rpm rfc2544-benchmarking tests test-name lty-test source-mac-address
```

```
00:00:5e:00:53:11
set services rpm rfc2544-benchmarking tests test-name lty-test destination-mac-address
00:00:5e:00:53:22
set services rpm rfc2544-benchmarking tests test-name lty-test ovlan-id 400
set services rpm rfc2544-benchmarking tests test-name lty-test service-type elan
set services rpm rfc2544-benchmarking tests test-name lty-test mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name lty-test family bridge
set services rpm rfc2544-benchmarking tests test-name lty-test direction egress
set services rpm rfc2544-benchmarking tests test-name lty-test source-udp-port 200
set services rpm rfc2544-benchmarking tests test-name lty-test destination-udp-port 200
set services rpm rfc2544-benchmarking tests test-name lty-test test-iterator-duration 20
set services rpm rfc2544-benchmarking tests test-name lty-test test-interface ge-1/1/3.0
set services rpm rfc2544-benchmarking tests test-name frloss-test test-profile frloss
set services rpm rfc2544-benchmarking tests test-name frloss-test source-mac-address
00:00:5e:00:53:11
set services rpm rfc2544-benchmarking tests test-name frloss-test destination-mac-address
00:00:5e:00:53:22
set services rpm rfc2544-benchmarking tests test-name frloss-test ovlan-id 400
set services rpm rfc2544-benchmarking tests test-name frloss-test service-type elan
set services rpm rfc2544-benchmarking tests test-name frloss-test mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name frloss-test family bridge
set services rpm rfc2544-benchmarking tests test-name frloss-test direction egress
set services rpm rfc2544-benchmarking tests test-name frloss-test source-udp-port 200
set services rpm rfc2544-benchmarking tests test-name frloss-test destination-udp-port 200
set services rpm rfc2544-benchmarking tests test-name frloss-test test-iterator-duration 20
set services rpm rfc2544-benchmarking tests test-name frloss-test test-interface ge-1/1/3.0
set interfaces ge-1/2/1 flexible-vlan-tagging
set interfaces ge-1/2/1 mtu 9192
set interfaces ge-1/2/1 encapsulation flexible-ethernet-services
set interfaces ge-1/2/1 unit 0 encapsulation vlan-bridge
set interfaces ge-1/2/1 unit 0 vlan-id 400
set interfaces ge-1/1/3 flexible-vlan-tagging
set interfaces ge-1/1/3 mtu 9192
set interfaces ge-1/1/3 encapsulation flexible-ethernet-services
set interfaces ge-1/1/3 unit 0 encapsulation vlan-bridge
set interfaces ge-1/1/3 unit 0 vlan-id 400
set bridge-domains bd1 vlan-id 600
set bridge-domains bd1 interface ge-1/2/1.0
set bridge-domains bd1 interface ge-1/1/3.0
```

Configuring Benchmarking Test Parameters on the MX104 Router

```
set services rpm rfc2544-benchmarking tests test-name l2b-reflector source-mac-address
00:00:5e:00:53:11
set services rpm rfc2544-benchmarking tests test-name l2b-reflector destination-mac-address
00:00:5e:00:53:22
set services rpm rfc2544-benchmarking tests test-name l2b-reflector service-type elan
set services rpm rfc2544-benchmarking tests test-name l2b-reflector mode reflect
set services rpm rfc2544-benchmarking tests test-name l2b-reflector family bridge
set services rpm rfc2544-benchmarking tests test-name l2b-reflector direction egress
set services rpm rfc2544-benchmarking tests test-name l2b-reflector test-interface ge-1/1/5.0
set interfaces ge-1/1/6 flexible-vlan-tagging
set interfaces ge-1/1/6 mtu 9192
set interfaces ge-1/1/6 encapsulation flexible-ethernet-services
set interfaces ge-1/1/6 unit 0 encapsulation vlan-bridge
set interfaces ge-1/1/6 unit 0 vlan-id 400
set interfaces ge-1/1/5 flexible-vlan-tagging
set interfaces ge-1/1/5 mtu 9192
set interfaces ge-1/1/5 encapsulation flexible-ethernet-services
set interfaces ge-1/1/5 unit 0 encapsulation vlan-bridge
set interfaces ge-1/1/5 unit 0 vlan-id 400
set bridge-domains bd1 domain-type bridge
set bridge-domains bd1 vlan-id 500
set bridge-domains bd1 interface ge-1/1/6.0
set bridge-domains bd1 interface ge-1/1/5.0
```

Configuring Throughput Benchmarking Test Parameters on the ACX Series Router

Step-by-Step Procedure

The following configuration requires you to configure a test profile for the throughput test and reference the test-profile in a unique test-name. The test-name defines the parameters for the throughput test to be performed on the ACX Series router.

To configure the throughput test parameters on the ACX Series router:

1. In configuration mode, at the [edit] hierarchy level, configure a real-time performance monitoring service (RPM) instance and an RFC2544-based benchmarking test for the RPM instance.

   ```
   [edit]
   user@host# edit services rpm rfc2544-benchmarking
   ```

2. Define a name for the first test profile—for example, `tput` for the throughput test profile.

```
[edit services rpm rfc2544-benchmarking]
user@host# set profiles test-profile tput
```

3. Configure the type of test to be performed as throughput, specify the packet size as 128 bytes, and define the theoretical maximum bandwidth for the test in kilobits per second (Kbps), with a value from 1 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles test-profile tput]
user@host# set test-type throughput packet-size 128 bandwidth-kbps 900000
```

4. Enter the `up` command twice to go to the [`edit services rpm rfc2544-benchmarking`] level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles test-profile tput ]
user@host# up
user@host# up
```

5. Define a name for the throughput test—for example, `tput-test`. The test name can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking ]
user@host# set tests test-name tput-test
```

6. Specify the name of the test profile, `tput`, to be associated with the test name.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set test-profile tput
```

7. Configure the source and destination MAC address for the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set source-mac-address 00:00:5e:00:53:11 destination-mac-address
00:00:5e:00:53:22
```

8. Configure the outer VLAN ID for the test frames and specify the service type under test to be E-LAN.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set ovlan-id 400 service-type elan
```

9. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set mode initiate-and-terminate
```

10. Configure the family type, `bridge`, for the benchmarking test and specify the direction, egress. Also, specify the source and destination UDP port to be used in the UDP headers of the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set family bridge direction egress source-udp-port 200 destination-udp-port 200
```

11. Specify the duration of each iteration in seconds, with a value from 10 seconds to 1,728,000 seconds, and specify the logical interface, ge-0/2/1.0, on which the RFC2544-benchmarking tests are run.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set test-iterator-duration 20 test-interface ge-1/1/3.0
```

Configuring Back-to-Back Frames Benchmarking Test Parameters on the ACX Series Router

Step-by-Step Procedure

The following configuration requires you to configure a test profile for the back to back frames test and reference the test-profile in a unique test-name. The test-name defines the parameters for the back to back frames test to be performed on the ACX Series router.

To configure the back-to-back frames test parameters on the ACX Series router:

1. In configuration mode, at the [edit] hierarchy level, configure a real-time performance monitoring service (RPM) instance and an RFC2544-based benchmarking test for the RPM instance.

```
[edit]
user@host# edit services rpm rfc2544-benchmarking
```

2. Define a name for the back-to-back test profile—for example, b2bt.

```
[edit services rpm rfc2544-benchmarking]
user@host# set profiles test-profile b2bt
```

3. Configure the type of test to be performed as back-to-back frames, specify the packet size as 128 bytes, and define the theoretical maximum bandwidth for the test in kilobits per second, with a value from 1 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles test-profile b2bt]
user@host# set test-type back-toback-frames packet-size 4444 bandwidth-kbps 950000
```

4. Enter the up command twice to go to the [edit services rpm rfc2544-benchmarking] level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles test-profile b2bt ]
user@host# up
user@host# up
```

5. Define a name for the back-to-back frames test—for example, b2bt-test. The test name can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking ]
user@host# set tests test-name b2bt-test
```

6. Specify the name of the test profile, b2bt, to be associated with the test name.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set test-profile b2bt
```

7. Configure the source and destination MAC address for the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set source-mac-address 00:00:5e:00:53:11 destination-mac-address
00:00:5e:00:53:22
```

8. Configure the outer VLAN ID for the test frames and specify the service type under test.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set ovlan-id 400 service-type elan
```

9. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set mode initiate-and-terminate
```

10. Configure the family type, `bridge`, for the benchmarking test and specify the direction, egress.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set family bridge direction egress
```

11. Specify the duration of each iteration in seconds, with a value from 10 seconds to 1,728,000 seconds. Also, specify the logical interface, ge-0/2/1.0, on which the RFC2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set test-iterator-duration 20 test-interface ge-1/1/3.0
```

Configuring Latency Benchmarking Test Parameters on the ACX Series Router

Step-by-Step Procedure

The following configuration requires you to configure a test profile for the latency test and reference the test-profile in a unique test-name. The test-name defines the parameters for the latency test to be performed on the ACX Series router.

To configure the latency test parameters on the ACX Series router:

1. In configuration mode, at the `[edit]` hierarchy level, configure a real-time performance monitoring service (RPM) instance and an RFC2544-based benchmarking test for the RPM instance.

   ```
   [edit]
   user@host# edit services rpm rfc2544-benchmarking
   ```

2. Define a name for the latency test profile—for example, `lty`.

   ```
   [edit services rpm rfc2544-benchmarking]
   user@host# set profiles test-profile lty
   ```

3. Configure the type of test to be performed as latency, specify the packet size of the test packet, and define the maximum bandwidth for the test in kilobits per second, with a value form 1 Kbps through 1,000,000 Kbps.

   ```
   [edit services rpm rfc2544-benchmarking profiles]
   user@host# set test-profile lty test-type latency packet-size 512 bandwidth-kbps 1000000
   ```

4. Enter the `up` command twice to go to the previous level in the configuration hierarchy.

   ```
   [edit services rpm rfc2544-benchmarking profiles test-profile lty]
   user@host# up
   user@host# up
   ```

5. Define a name for the latency test—for example, `lty-test`. The test name can be up to 32 characters in length.

   ```
   [edit services rpm rfc2544-benchmarking ]
   user@host# set tests test-name lty-test
   ```

6. Specify the name of the test profile, `lty`, to be associated with the test name.

   ```
   [edit services rpm rfc2544-benchmarking tests test-name lty-test]
   user@host# set test-profile lty
   ```

7. Configure the source and destination MAC address for the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set source-mac-address 00:00:5e:00:53:11 destination-mac-address
00:00:5e:00:53:22
```

8. Configure the outer VLAN ID for the test frames and specify the service type under test.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set ovlan-id 400 service-type elan
```

9. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set mode initiate-and-terminate
```

10. Configure the family type, `bridge`, for the benchmarking test and specify the direction, egress. Also, specify the source and destination UDP port to be used in the UDP headers of the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set family bridge direction egress source-udp-port 200 destination-udp-port 200
```

11. Specify the duration of each iteration in seconds, with a value from 10 seconds to 1,728,000 seconds. Also, specify the logical interface, ge-0/2/1.0, on which the RFC2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set test-iterator-duration 20 test-interface ge-1/1/3.0
```

Configuring Frame Loss Benchmarking Test Parameters on the ACX Series Router

Step-by-Step Procedure

The following configuration requires you to configure a test profile for the frame loss test and reference the test-profile in a unique test-name. The test-name defines the parameters for the frame loss test to be performed on the ACX Series router.

To configure the frame loss test parameters on the ACX Series router:

1. In configuration mode, at the [edit] hierarchy level, configure a real-time performance monitoring service (RPM) instance and an RFC2544-based benchmarking test for the RPM instance.

```
[edit]
user@host# edit services rpm rfc2544-benchmarking
```

2. Define a name for the frame loss test profile—for example, frloss.

```
[edit services rpm rfc2544-benchmarking]
user@host# set profiles test-profile frloss
```

3. Configure the type of test performed as frame loss, specify the packet size of the test packet, and define the maximum bandwidth for the test in kilobits per second, with a value from 1 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles]
user@host# set test-profile frloss test-type frame-loss packet-size 1600 bandwidth-kbps
1000000
```

4. Enter the up command to go to the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles ]
user@host# up
```

5. Define a name for the frame loss test—for example, frloss-test. The test name can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking ]
user@host# set tests test-name frloss-test
```

6. Specify the name of the test profile, frloss, to be associated with the test name.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set test-profile frloss
```

7. Configure the source and destination MAC address for the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set source-mac-address 00:00:5e:00:53:11 destination-mac-address
00:00:5e:00:53:22
```

8. Configure the outer VLAN ID for the test frames and specify the service type under test.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set ovlan-id 400 service-type elan
```

9. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set mode initiate-and-terminate
```

10. Configure the family type, `bridge`, for the benchmarking test and specify the direction, egress. Also, specify the source and destination UDP port to be used in the UDP headers of the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set family bridge direction egress source-udp-port 200 destination-udp-port 200
```

11. Specify the duration of each iteration in seconds, with a value from 10 seconds to 1,728,000 seconds. Also, specify the logical interface, ge-0/2/1.0, on which the RFC2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set test-iterator-duration 20 test-interface ge-1/1/3.0
```

12. Enter the `exit` command to go to the [edit] hierarchy level.

```
[edit services rpm rfc2544-benchmarking tests test-name test4 ]
user@host# exit
```

Configuring Other Benchmarking Test Parameters on the ACX Series Router

Step-by-Step Procedure

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

To configure the interface and bridge domain on the ACX Series router:

1. Configure the Layer 2 NNI interface on which the tests must be run from the [edit] hierarchy level.

```
[edit]
user@host# edit interfaces ge-1/2/1
```

2. Configure flexible VLAN tagging for the transmission of untagged frames or 802.1Q single-tagged and dual-tagged frames on the logical interface. You can also specify the maximum transmission unit (MTU) size for the interface and the encapsulation.

```
[edit interfaces ge-1/2/1]
user@host# set flexible-vlan-tagging mtu 9192 encapsulation flexible-ethernet-services
```

3. Configure a logical unit for the interface, specify the encapsulation, and configure the VLAN ID on the logical interfaces.

```
[edit interfaces ge-1/2/1]
user@host# set unit 0 encapsulation vlan-bridge vlan-id 400
```

4. Configure the Layer 2 UNI interface.

```
[edit]
user@host# edit interfaces ge-1/1/3
```

5. Configure flexible VLAN tagging for transmission of non-tagged frames or 802.1Q single-tag and dual-tag frames on the logical interface. You can also specify the maximum transmission unit (MTU) size for the interface and the encapsulation.

```
[edit interfaces ge-1/1/3]
user@host# set flexible-vlan-tagging mtu 9192 encapsulation flexible-ethernet-services
```

6. Configure a logical unit for the interface and specify the encapsulation and configure the VLAN ID on the logical interfaces.

```
[edit interfaces ge-1/1/3]
user@host# set unit 0 encapsulation vlan-bridge vlan-id 400
```

7. Configure the bridge domain, bd1, and specify the VLAN ID associated with the bridge domain and the associated interfaces from the [edit] hierarchy level.

```
[edit]
user@host# set bridge-domains bd1 vlan-id 600 interface ge-1/2/1.0
user@host# set bridge-domains bd1 vlan-id 600 intreface ge-1/1/3.0
```

Configuring Benchmarking Test Parameters on the MX104 Router

Step-by-Step Procedure

The following configuration requires you to configure a unique test-name for the benchmarking test on the MX104 Series router. The test-name defines the parameters for the benchmarking test to be performed. Because the test interface and test MAC addresses are the same, you can create a single test configuration at the reflector (MX104).

To configure the benchmarking test parameters on the MX104 Series router:

1. In configuration mode, at the [edit] hierarchy level, configure a real-time performance monitoring service (RPM) instance and an RFC2544-based benchmarking test for the RPM instance.

```
[edit]
user@host# edit services rpm rfc2544-benchmarking
```

2. Define a name for the test—for example, l2b-reflector. The test name can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking ]
user@host# set tests test-name l2b-reflector
```

3. Specify the source and destination MAC addresses of the test packet.

```
[edit services rpm rfc2544-benchmarking test-name l2b-reflector]
user@host# set source-mac-address 00:00:5e:00:53:11 destination-mac-address 00:00:5e:00:53:22
```

4. Specify the service type under test and the mode, which is `reflect`, at the reflector.

```
[edit services rpm rfc2544-benchmarking test-name l2b-reflector]
user@host# set service-type elan
```

5. Specify the mode which is reflect at the reflector.

```
[edit services rpm rfc2544-benchmarking test-name l2b-reflector]
user@host# set mode reflect
```

6. Configure the family type, `bridge`, and specify the direction, egress, for the benchmarking test. Also, specify the logical interface, ge-1/1/5.0, on which the RFC2544-based benchmarking test is being run.

```
[edit services rpm rfc2544-benchmarking tests test-name l2b-reflector]
user@host# set family bridge direction egress test-interface ge-1/1/5.0
```

Configuring Other Benchmarking Test Parameters on the MX104 Router

Step-by-Step Procedure

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

To configure the interface and bridge domain on the MX104 Series router:

1. Configure the Layer 2 NNI interface on which the tests must be run.

```
[edit]
user@host# edit interfaces ge-1/1/6
```

2. Configure flexible VLAN tagging for transmission of untagged frames or 802.1Q single-tagged and dual-tagged frames on the logical interface. You can also specify the maximum transmission unit (MTU) size for the interface and the encapsulation.

```
[edit interfaces ge-1/1/6]
user@host# set flexible-vlan-tagging mtu 9192 encapsulation flexible-ethernet-services
```

3. Configure a logical unit for the interface, specify the encapsulation, and configure the VLAN ID on the logical interface.

```
[edit interfaces ge-1/1/6]
user@host# set unit 0 encapsulation vlan-bridge vlan-id 400
```

4. Configure the Layer 2 NNI interface.

```
[edit]
user@host# edit interfaces ge-1/1/5
```

5. Configure flexible VLAN tagging for transmission of untagged frames or 802.1Q single-tagged and dual-tagged frames on the logical interface. You can also specify the maximum transmission unit (MTU) size for the interface and the encapsulation.

```
[edit interfaces ge-1/1/5]
user@host# set flexible-vlan-tagging mtu 9192 encapsulation flexible-ethernet-services
```

6. Configure a logical unit for the interface, specify the encapsulation, and configure the VLAN ID on the logical interfaces.

```
[edit interfaces ge-1/1/5]
user@host# set unit 0 encapsulation vlan-bridge vlan-id 400
```

7. Configure the bridge domain, bd1, and specify the VLAN ID associated with the bridge domain, and the associated interfaces from the [edit] hierarchy level.

```
[edit]
user@host# set bridge-domains bd1 vlan-id 500 interface ge-1/1/6.0
user@host# set bridge-domains bd1 vlan-id 500 interface ge-1/1/5.0
```

8. Start the benchmarking test on the reflector.

```
user@host> test services rpm rfc2544-benchamrking test l2b-reflector start
```

After the test is successfully completed at the initiator, you can stop the test at the reflector by entering the `test services rpm rfc2544-benchmarking test l2b-reflector stop` command.

Results

In configuration mode, confirm your configuration on the ACX Series router and the MX104 Series router by entering the `show` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

Benchmarking Test Parameters on the ACX Series router :

```
[edit interfaces]
  ge-1/2/1 {
      flexible-vlan-tagging;
      mtu 9192;
      encapsulation flexible-ethernet-services;
      unit 0 {
          encapsulation vlan-bridge;
          vlan-id 400;
      }
  }
  ge-1/1/3 {
      flexible-vlan-tagging;
      mtu 9192;
      encapsulation flexible-ethernet-services;
      unit 0 {
          encapsulation vlan-bridge;
          vlan-id 400;
      }
  }
```

```
[edit bridge-domains]
 bd1 {
    vlan-id 600;
    interface ge-1/2/1.0;
    interface ge-1/1/3.0;
 }

[edit services rpm]
  rfc2544-benchmarking {
      profiles {
          test-profile tput {
              test-type throughput
              packet-size 128;
              bandwidth-kbps 900000;
          }
          test-profile b2bt {
              test-type back-back-frames
              packet-size 512;
              bandwidth-kbps 950000;
          }
          test-profile lty {
              test-type latency
              packet-size 512;
              bandwidth-kbps 100000;
          }
          test-profile frloss {
              test-type frameloss
              packet-size 1600;
              bandwidth-kbps 1000000;
          }
      }
      tests {
          test-name tput-test {
              interface ge-1/1/3.0;
              test-profile tput;
              mode initiate-and-terminate;
              source-mac-address 00:00:5e:00:53:11;
              destination-mac-address 00:00:5e:00:53:22;
              ovlan-id 400;
              service-type elan;
              family bridge;
              direction egress;
```

```
        source-udp-port 200;
        destination-udp-port 200;
        test-iterator-duration 20;
    }
    test-name b2b-test {
        interface ge-1/1/3.0;
        test-profile b2bt;
        mode initiate-and-terminate;
        source-mac-address 00:00:5e:00:53:11;
        destination-mac-address 00:00:5e:00:53:22;
        ovlan-id 400;
        service-type elan;
        family bridge;
        direction egress;
        test-iterator-duration 20;
    }
    test-name lty-test {
        interface ge-1/1/3.0;
        test-profile lty;
        mode initiate-and-terminate;
        source-mac-address 00:00:5e:00:53:11;
        destination-mac-address 00:00:5e:00:53:22;
        ovlan-id 400;
        service-type elan;
        family bridge;
        direction egress;
        source-udp-port 200;
        destination-udp-port 200;
        test-iterator-duration 20;
    }
    test-name frloss-test {
        interface ge-1/1/3.0;
        test-profile frloss;
        mode initiate-and-terminate;
        source-mac-address 00:00:5e:00:53:11;
        destination-mac-address 00:00:5e:00:53:22;
        ovlan-id 400;
        service-type elan;
        family bridge;
        direction egress;
        source-udp-port 200;
        destination-udp-port 200;
        test-iterator-duration 20;
```

```
        }
    }
  }
```

Benchmarking Test Parameters on the MX104 Series router:

```
[edit interfaces]
  ge-1/1/6 {
      flexible-vlan-tagging;
      mtu 9192;
      encapsulation flexible-ethernet-services;
      unit 0 {
          encapsulation vlan-bridge;
          vlan-id 400;
      }
  }

 ge-1/1/5 {
      flexible-vlan-tagging;
      mtu 9192;
      encapsulation flexible-ethernet-services;
      unit 0 {
          encapsulation vlan-bridge;
          vlan-id 400;
      }
 }
}
[edit bridge-domains]
 bd1 {
    vlan-id 500;
    interface ge-1/1/6.0;
    interface ge-1/1/5.0;
 }
[edit services rpm]
  rfc2544-benchmarking {
      # Note, When in reflector mode, test profile is not needed
      tests {
          test-name l2b-reflector {
              interface ge-1/1/5.0;
              source-mac-address 00:00:5e:00:53:11;
              destination-mac-address 00:00:5e:00:53:22;
              family bridge;
```

```
            mode reflect;
            service-type elan;
            family bridge;
            direction egress;
        }
    }
}
```

Verifying the Results of the Benchmarking Tests for Layer 2 Services (E-LAN) in Bridge Domains

IN THIS SECTION

Examine the results of the benchmarking tests that are performed on the configured service between the ACX Series router and the MX104 Series router. Start the test on the reflector first and then start the test on the initiator.

Verifying the Throughput Benchmarking Test Results

Purpose

Verify that the necessary and statistical values are displayed for the benchmarking tests that are run on the configured service between the ACX Series router and the MX104 Series router.

Action

In operational mode, enter the `show services rpm rfc2544-benchmarking test-id` *`test-id-number`* `detail` command on the ACX Series router.

```
user@host> show services rpm rfc2544-benchmarking test-id 1 detail
Test information :
      Test id: 1, Test name: tput_test, Test type: Throughput
```

```
      Test mode: Initiate-and-Terminate
      Test packet size: 128
      Test state: TEST_STATE_COMPLETED
      Status: Test-Completed
      Test start time: 2014-09-24 22:21:09 PDT
      Test finish time: 2014-09-24 22:21:33 PDT
      Counters last cleared: Never

      Test-profile Configuration:
          Test-profile name: tput
          Test packet size: 128
          Theoretical max bandwidth : 900000 kbps

      Test Configuration:
          Test mode: Initiate-and-Terminate
          Duration in seconds: 20
          Test finish wait duration in seconds: 1
          Test family: Bridge
          Test iterator pass threshold: 0.50 %
          Test receive failure threshold: 0.00 %
          Test transmit failure threshold: 0.50 %

      Bridge family Configuration:
          Interface : ge-1/1/3.0
          Test direction: Egress
          Source mac address: 00:00:5e:00:53:11
          Destination mac address: 00:00:5e:00:53:22
          Outer vlan-id: 400
          Outer vlan priority: 0
          Outer vlan cfi: 0
          Outer tag protocol id: 0x8100
          Source ipv4 address: 192.168.1.10
          Destination ipv4 address: 192.168.1.20
          Source udp port: 200
          Destination udp port: 200

  Rfc2544 throughput test information :
      Initial test load percentage : 100.00 %
      Test iteration mode : Binary
      Test iteration step : 50.00 %
      Theoretical max bandwidth : 900000 kbps

Test packet size: 128
```

```
Iteration  Internal Duration  Elapsed  --------  Throughput ---------
           Overhead  (sec)     time    Theoretical  Transmit Measured
   1         0       20        20       100.00 %    100.00 % 100.00 %

Result of the iteration runs : Throughput Test complete for packet size 128
Best iteration : 1, Best iteration (pps) : 760135
Best iteration throughput : 100.00 %

RFC2544 Throughput test results summary:
----------------------------------------


  Packet  Internal    Theoretical Transmit    Tx           Rx       Measured        Measured
   Size   overhead    rate (pps)    pps     Packets      Packets   throughput %   bandwidth
(kbps)
   128     0           760135     760135    15202700     15202700    100.00 %       900000
```

In operational mode, enter the `show services rpm rfc2544-benchmarking test-id` *`test-id-number`* `detail` command on the MX104 Series router.

```
user@host> show services rpm rfc2544-benchmarking test-id 1 detail
Test information :
      Test id: 1, Test name: l2b-reflector, Test type: Reflect
      Test mode: Reflect
      Test packet size: 0
      Test state: TEST_STATE_RUNNING
      Status: Running
      Test start time: 2014-09-24 22:20:54 PDT
      Test finish time: TEST_RUNNING
      Counters last cleared: Never

      Test Configuration:
          Test mode: Reflect
          Duration in seconds: 864000
          Test finish wait duration in seconds: 1
          Test family: Bridge
          Test iterator pass threshold: 0.50 %
          Test receive failure threshold: 0.00 %
          Test transmit failure threshold: 0.50 %

      Bridge family Configuration:
          Interface : ge-1/1/5.0
          Test direction: Egress
```

```
        Source mac address: 00:00:5e:00:53:11
        Destination mac address: 00:00:5e:00:53:22
        Service type: Elan

Elapsed      Reflected          Reflected
 time         Packets             Bytes
    61       15202700          1945945600
```

You can also use the `show services rpm rfc2544-benchmarking (aborted-test | active-tests | completed-tests | summary)` command to display information about the results of each category or state of the RFC2544-based benchmarking tests for each real-time performance monitoring (RPM) instance.

Meaning

The output displays the details of the benchmarking test that was performed. For more information about the `run show services rpm rfc2544-benchmarking` operational command, see `show services rpm rfc2544-benchmarking` in the CLI Explorer.

Verifying the Back-to-Back Benchmarking Test Results

Purpose

Verify that the necessary and statistical values are displayed for the benchmarking tests that are run on the configured service between the ACX Series router and the MX104 Series router.

Action

In operational mode, enter the `show services rpm rfc2544-benchmarking test-id` *`test-id-number`* `detail` command on the ACX Series router.

```
user@host> show services rpm rfc2544-benchmarking test-id 4 detail
Test information :
      Test id: 4, Test name: b2b-test, Test type: Back-Back-Frames
      Test mode: Initiate-and-Terminate
      Test packet size: 128 512
      Test state: TEST_STATE_COMPLETED
      Status: Test-Completed
      Test start time: 2014-09-24 22:30:16 PDT
      Test finish time: 2014-09-24 22:31:03 PDT
      Counters last cleared: Never
```

```
    Test-profile Configuration:
        Test-profile name: b2bt
        Test packet size: 128 512
        Theoretical max bandwidth : 950000 kbps

    Test Configuration:
        Test mode: Initiate-and-Terminate
        Duration in seconds: 20
        Test finish wait duration in seconds: 1
        Test family: Bridge
        Test iterator pass threshold: 0.50 %
        Test receive failure threshold: 0.00 %
        Test transmit failure threshold: 0.50 %

    Bridge family Configuration:
        Interface : ge-1/1/3.0
        Test direction: Egress
        Source mac address: 00:00:5e:00:53:11
        Destination mac address: 00:00:5e:00:53:22
        Outer vlan-id: 400
        Outer vlan priority: 0
        Outer vlan cfi: 0
        Outer tag protocol id: 0x8100
        Source ipv4 address: 192.168.1.10
        Destination ipv4 address: 192.168.1.20
        Source udp port: 4040
        Destination udp port: 4041

  Rfc2544 Back-Back test information :
    Initial burst length: 20 seconds at 950000 kbps
    Test iteration mode : Binary
    Test iteration step : 50.00 %

Test packet size: 128
Iteration   Theoretical    Transmit     Internal Duration  Elapsed
            burst length  burst length  overhead   time
             (packets)     (packets)
    1    16047280        16047280       0       20        20

Result of the iteration runs : Back-Back Test complete for packet size 128
Best iteration : 1
Measured burst (num sec)  : 20 sec
Measured burst (num pkts)  : 16047280 packets
```

```
Test packet size: 512
Iteration   Theoretical    Transmit     Internal Duration  Elapsed
            burst length  burst length  overhead   time
             (packets)     (packets)
    1    4464280         4464280        0       20        20

Result of the iteration runs : Back-Back Test complete for packet size 512
Best iteration : 1
Measured burst (num sec)  : 20 sec
Measured burst (num pkts)  : 4464280 packets

RFC2544 Back-Back test results summary:
----------------------------------------


  Packet      Measured Burst      Time
   Size      length (Packets)   (seconds)
    128          16047280        20
    512           4464280        20
```

In operational mode, enter the `show services rpm rfc2544-benchmarking test-id` *`test-id-number`* `detail` command on the MX104 Series router.

```
user@host> show services rpm rfc2544-benchmarking test-id 4 detail
Test information :
      Test id: 4, Test name: l2b-reflector, Test type: Reflect
      Test mode: Reflect
      Test packet size: 0
      Test state: TEST_STATE_RUNNING
      Status: Running
      Test start time: 2014-09-24 22:30:07 PDT
      Test finish time: TEST_RUNNING
      Counters last cleared: Never

      Test Configuration:
          Test mode: Reflect
          Duration in seconds: 864000
          Test finish wait duration in seconds: 1
          Test family: Bridge
          Test iterator pass threshold: 0.50 %
          Test receive failure threshold: 0.00 %
          Test transmit failure threshold: 0.50 %
```

```
      Bridge family Configuration:
          Interface : ge-1/1/5.0
          Test direction: Egress
          Source mac address: 00:00:5e:00:53:11
          Destination mac address: 00:00:5e:00:53:22
          Service type: Elan

Elapsed        Reflected           Reflected
 time           Packets              Bytes
     58        20511560           4339763200
```

You can also use the `show services rpm rfc2544-benchmarking (aborted-test | active-tests | completed-tests | summary)` command to display information about the results of each category or state of the RFC2544-based benchmarking tests for each real-time performance monitoring (RPM) instance.

Meaning

The output displays the details of the benchmarking test that was performed. For more information about the `run show services rpm rfc2544-benchmarking` operational command, see `show services rpm rfc2544-benchmarking` in the CLI Explorer.

Verifying the Frame Loss Benchmarking Test Results

Purpose

Verify that the necessary and statistical values are displayed for the benchmarking tests that are run on the configured service between the ACX Series router and the MX104 Series router.

Action

In operational mode, enter the `show services rpm rfc2544-benchmarking test-id` *`test-id-number`* `detail` command on the ACX Series router.

```
user@host> show services rpm rfc2544-benchmarking test-id 3 detail
Test information :
      Test id: 3, Test name: frloss-test, Test type: Frame-Loss
      Test mode: Initiate-and-Terminate
      Test packet size: 1600
      Test state: TEST_STATE_COMPLETED
      Status: Test-Completed
```

```
      Test start time: 2014-09-24 22:26:45 PDT
      Test finish time: 2014-09-24 22:27:55 PDT
      Counters last cleared: Never

      Test-profile Configuration:
          Test-profile name: frloss
          Test packet size: 1600
          Theoretical max bandwidth : 1000000 kbps

      Test Configuration:
          Test mode: Initiate-and-Terminate
          Duration in seconds: 20
          Test finish wait duration in seconds: 1
          Test family: Bridge
          Test iterator pass threshold: 0.50 %
          Test receive failure threshold: 0.00 %
          Test transmit failure threshold: 0.50 %

      Bridge family Configuration:
          Interface : ge-1/1/3.0
          Test direction: Egress
          Source mac address: 00:00:5e:00:53:11
          Destination mac address: 00:00:5e:00:53:22
          Outer vlan-id: 400
          Outer vlan priority: 0
          Outer vlan cfi: 0
          Outer tag protocol id: 0x8100
          Source ipv4 address: 192.168.1.10
          Destination ipv4 address: 192.168.1.20
          Source udp port: 200
          Destination udp port: 200

  Rfc2544 frame-loss test information :
      Initial test load percentage : 100.00 %
      Test iteration mode : step-down
      Test iteration step : 10 %
      Theoretical max bandwidth : 1000000 kbps

Test packet size: 1600
Iteration Internal Duration  Elapsed  --------  Throughput ---------  Frame-loss
          Overhead  (sec)     time    Theoretical  Transmit Measured    rate %
   1        0        20       20       100.00 %     100.00 % 100.00 %  0.00 %
   2        0        20       20       100.00 %     100.00 % 100.00 %  0.00 %
```

```
   3        0        20       20       100.00 %     100.00 % 100.00 %  0.00 %

Result of the iteration runs : Frame-loss test complete for packet size 1600
Percentage throughput transmitted: 100.00 %
Frame-loss rate (percent) : 0.00 %

RFC2544 Frame-loss test results summary:
----------------------------------------


  Packet  Internal  Theoretical  Transmit    Transmit      Tx             Rx            Frame
Loss
   Size   overhead  rate (pps)      pps     throughput    Packets        Packets       rate
percent
    1600   0         77160       77160      100.00 %      1543200        1543200        0.00 %
```

In operational mode, enter the `show services rpm rfc2544-benchmarking test-id` *`test-id-number`* `detail` command on the MX104 Series router.

```
user@host> show services rpm rfc2544-benchmarking test-id 3 detail
Test information :
      Test id: 3, Test name: l2b-reflector, Test type: Reflect
      Test mode: Reflect
      Test packet size: 0
      Test state: TEST_STATE_RUNNING
      Status: Running
      Test start time: 2014-09-24 22:25:36 PDT
      Test finish time: TEST_RUNNING
      Counters last cleared: Never

      Test Configuration:
          Test mode: Reflect
          Duration in seconds: 864000
          Test finish wait duration in seconds: 1
          Test family: Bridge
          Test iterator pass threshold: 0.50 %
          Test receive failure threshold: 0.00 %
          Test transmit failure threshold: 0.50 %

      Bridge family Configuration:
          Interface : ge-1/1/5.0
          Test direction: Egress
          Source mac address: 00:00:5e:00:53:11
```

```
          Destination mac address: 00:00:5e:00:53:22
          Service type: Elan

Elapsed        Reflected            Reflected
 time           Packets               Bytes
     95         1624361            2598977600
```

You can also use the `show services rpm rfc2544-benchmarking (aborted-test | active-tests | completed-tests | summary)` command to display information about the results of each category or state of the RFC2544-based benchmarking tests for each real-time performance monitoring (RPM) instance.

Meaning

The output displays the details of the benchmarking test that was performed. For more information about the `run show services rpm rfc2544-benchmarking` operational command, see `show services rpm rfc2544-benchmarking` in the CLI Explorer.

Verifying the Latency Benchmarking Test Results

Purpose

Verify that the necessary and statistical values are displayed for the benchmarking tests that are run on the configured service between the ACX Series router and the MX104 Series router.

Action

In operational mode, enter the `show services rpm rfc2544-benchmarking test-id` *`test-id-number`* `detail` command on the ACX Series router.

```
user@host> show services rpm rfc2544-benchmarking test-id 5 detail
Test information :
      Test id: 5, Test name: lty-test, Test type: Latency
      Test mode: Initiate-and-Terminate
      Test packet size: 512
      Test state: TEST_STATE_COMPLETED
      Status: Test-Completed
      Test start time: 2014-09-24 22:33:05 PDT
      Test finish time: 2014-09-24 22:40:46 PDT
      Counters last cleared: Never

      Test-profile Configuration:
```

```
            Test-profile name: lty
            Test packet size: 512
            Theoretical max bandwidth : 1000000 kbps

        Test Configuration:
            Test mode: Initiate-and-Terminate
            Duration in seconds: 20
            Test finish wait duration in seconds: 1
            Test family: Bridge
            Test iterator pass threshold: 0.50 %
            Test receive failure threshold: 0.00 %
            Test transmit failure threshold: 0.50 %

        Bridge family Configuration:
            Interface : ge-1/1/3.0
            Test direction: Egress
            Source mac address: 00:00:5e:00:53:11
            Destination mac address: 00:00:5e:00:53:22
            Outer vlan-id: 400
            Outer vlan priority: 0
            Outer vlan cfi: 0
            Outer tag protocol id: 0x8100
            Source ipv4 address: 192.168.1.10
            Destination ipv4 address: 192.168.1.20
            Source udp port: 200
            Destination udp port: 200

    Rfc2544 latency test information :
        Theoretical max bandwidth : 1000000 kbps
        Initial test load percentage : 100.00 %
        Duration in seconds: 20
        Measurement unit for timestamp: Nanoseconds

  Test packet size: 512
  Iteration  Duration   Elapsed  Theoretical   Transmit      Throughput    ------------------
  Latency --------------------
              (sec)      time    rate (pps)       pps      percent          Minimum
  Average      Maximum     Probe
     1        20         20      234962         234962        100.00 %        44008
  45253        47424       45096
     2        20         20      234962         234962        100.00 %        44008
  45237        47456       45256
     3        20         20      234962         234962        100.00 %        43864
```

```
45198       46976      45144
   4        20         20      234962         234962       100.00 %        43832
45243       47088      45096
   5        20         20      234962         234962       100.00 %        44072
45261       46976      45176
   6        20         20      234962         234962       100.00 %        43784
45214       46864      45032
   7        20         20      234962         234962       100.00 %        44024
45259       47216      45240
   8        20         20      234962         234962       100.00 %        44072
45290       46864      45192
   9        20         20      234962         234962       100.00 %        43976
45272       46792      45208
  10        20         20      234962         234962       100.00 %        44024
45206       46976      45112
  11        20         20      234962         234962       100.00 %        44040
45198       47088      45176
  12        20         20      234962         234962       100.00 %        44008
45223       46976      45160
  13        20         20      234962         234962       100.00 %        44088
45257       47408      45176
  14        20         20      234962         234962       100.00 %        43976
45183       46832      45080
  15        20         20      234962         234962       100.00 %        44024
45198       47088      45112
  16        20         20      234962         234962       100.00 %        43864
45206       46912      45208
  17        20         20      234962         234962       100.00 %        44056
45209       46960      45176
  18        20         20      234962         234962       100.00 %        44008
45198       46912      45112
  19        20         20      234962         234962       100.00 %        43816
45175       47248      45000
  20        20         20      234962         234962       100.00 %        43912
45202       46992      45192

Result of the iteration runs : Latency Test complete for packet size 512
Internal overhead per packet: 0
Avg (min) Latency                  : 43972
Avg (avg) latency                  : 45224
Avg (Max) latency                  : 47052
Avg (probe) latency                : 45147
```

```
RFC2544 Latency test results summary:
----------------------------------------

  Packet Internal  Theoretical  Transmit     Tx           Rx         ------------------ Latency
--------------------
   Size  overhead  rate (pps)      pps     Packets       Packets      Minimum     Average
Maximum     Probe
   512     0        234962      234962      93984800      93984800     43972       45224
47052       45147
```

In operational mode, enter the `show services rpm rfc2544-benchmarking test-id` *`test-id-number`* `detail` command on the MX104 Series router.

```
user@host> show services rpm rfc2544-benchmarking test-id 5 detail
Test information :
      Test id: 5, Test name: l2b-reflector, Test type: Reflect
      Test mode: Reflect
      Test packet size: 0
      Test state: TEST_STATE_RUNNING
      Status: Running
      Test start time: 2014-09-24 22:32:55 PDT
      Test finish time: TEST_RUNNING
      Counters last cleared: Never

      Test Configuration:
          Test mode: Reflect
          Duration in seconds: 864000
          Test finish wait duration in seconds: 1
          Test family: Bridge
          Test iterator pass threshold: 0.50 %
          Test receive failure threshold: 0.00 %
          Test transmit failure threshold: 0.50 %

      Bridge family Configuration:
          Interface : ge-1/1/5.0
          Test direction: Egress
          Source mac address: 00:00:5e:00:53:11
          Destination mac address: 00:00:5e:00:53:22
          Service type: Elan

Elapsed        Reflected            Reflected
```

```
time          Packets          Bytes
   426        84586320         43308195840
```

You can also use the `show services rpm rfc2544-benchmarking (aborted-test | active-tests | completed-tests | summary)` command to display information about the results of each category or state of the RFC2544-based benchmarking tests for each real-time performance monitoring (RPM) instance.

Meaning

The output displays the details of the benchmarking test that was performed. For more information about the `show services rpm rfc2544-benchmarking` operational mode command, see `show services rpm rfc2544-benchmarking` topic in the CLI Explorer.

RELATED DOCUMENTATION

Understanding RFC2544-Based Benchmarking Tests for E-LAN and E-Line Services on MX Series Routers | 800

Supported RFC 2544-Based Benchmarking Statements on MX Series Routers | 804

Example: Configuring Benchmarking Tests to Measure SLA Parameters for E-LAN Services on an MX104 Router Using VPLS

IN THIS SECTION

This example shows how to configure benchmarking tests for the E-LAN services using BGP-based VPLS. The example covers the four benchmarking tests: throughput, frame loss, back-to-back frames, and latency.

Requirements

NOTE: MX Series routers support only the reflector function in RFC2544-based benchmarking tests.

This example uses the following hardware and software components:

- An MX104 3D Universal Edge Router (reflector)
- Any MX Series router
- Any ACX Series router (initiator)
- Junos OS Release 15.1 or later for MX Series routers

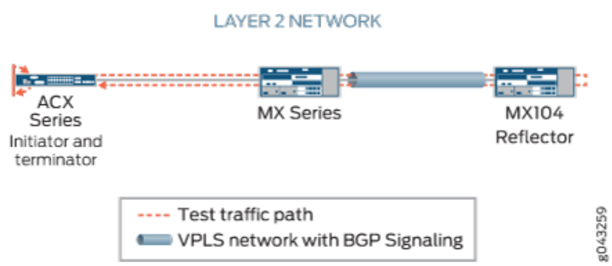
Overview

Consider a sample topology in which an ACX Series router functions as an initiator and terminator of the test frames for an RFC2544-based benchmarking test. The ACX Series router is connected to a provider edge router, PE1 (an MX Series router). The PE1 router is configured with a VPLS routing instance and is connected over a Layer 2 network to another provider edge router, PE2 (an MX104 Series router). A simple VPLS network with BGP signaling is created between routers PE1 and PE2. The MX104 Series router also functions as a reflector to reflect the test frames it receives from the ACX Series router back to the initiator.

Benchmarking tests compute the performance attributes in the user-to-network interface (UNI) direction of the Layer 2 E-LAN service between the ACX Series router and the MX104 Series router. To measure SLA parameters for E-LAN services using VPLS, configure specific benchmarking tests. In this example, all four benchmarking tests (throughput, back-to-back frames, latency, and frame-loss) are configured.

Figure 72 on page 892 shows the sample topology to perform all four RFC2544-based benchmarking tests for the UNI direction on a Layer 2 network using VPLS.

Figure 72: Layer 2 Reflection with Simple BGP-based VPLS Topology



On the ACX Series router, ge-0/2/1.0 is the Layer 2 NNI interface and ge-0/2/0.0 is the Layer 2 UNI interface. For each benchmarking test configured on the ACX Series router, specify the source MAC address as 00:00:5e:00:53:11 and 00:00:5e:00:53:22 as the destination MAC address. Also, specify the VLAN ID as 512. On the MX Series router, ge-0/3/0.0 is the Layer 2 NNI interface and ge-0/2/1.0 is the UNI interface. On the MX104 Series router, ge-0/2/5.0 is the Layer 2 NNI interface and ge-0/3/1.0 is the Layer 2 UNI interface. The benchmarking tests are used to compute the performance attributes for an E-LAN service using VPLS.

Configuration

IN THIS SECTION

In this example, you configure the benchmarking tests for the UNI direction for a Layer 2 E-LAN service using VPLS between two routers (initiator and reflector) to detect and analyze the performance of the interconnected routers. The initiator and reflector routers are not directly connected to each other. The initiator is connected to a provider edge router (PE1), which is in turn connected to the reflector. In this example, the ACX Series router is the initiator, an MX Series router is PE1, and the MX104 router is the other provider edge router (PE2) and reflector. Start by configuring the initiator. On the ACX Series router, you first configure each test by specifying the test profile and the test attributes, and then define the test by associating the test with the test profile with the relevant attributes. You can then configure the interface. On the MX Series router, configure the VPLS parameters to enable VPLS on the router. On the MX104 Series router, configure the benchmarking parameters and the VPLS parameters.

NOTE: When you configure Layer 2 reflection, you can specify the service type under test as ELINE if you want to simulate an Eline service by using bridge encapsulation.

CLI Quick Configuration

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

Configuring Benchmarking Test Parameters on the ACX Series Router (Initiator)

```
set services rpm rfc2544-benchmarking profiles test-profile tput test-type throughput
set services rpm rfc2544-benchmarking profiles test-profile tput packet-size 256
set services rpm rfc2544-benchmarking profiles test-profile tput bandwidth-kbps 600000
set services rpm rfc2544-benchmarking profiles test-profile b2bt test-type back-back-frames
set services rpm rfc2544-benchmarking profiles test-profile b2bt packet-size 9104
set services rpm rfc2544-benchmarking profiles test-profile b2bt bandwidth-kbps 600000
set services rpm rfc2544-benchmarking profiles test-profile lty test-type latency
set services rpm rfc2544-benchmarking profiles test-profile lty packet-size 1024
set services rpm rfc2544-benchmarking profiles test-profile lty bandwidth-kbps 6000000
set services rpm rfc2544-benchmarking profiles test-profile frloss test-type frame-loss
set services rpm rfc2544-benchmarking profiles test-profile frloss packet-size 1600
set services rpm rfc2544-benchmarking profiles test-profile frloss bandwidth-kbps 6000000
set services rpm rfc2544-benchmarking profiles test-profile frloss step-percent 5
set services rpm rfc2544-benchmarking tests test-name tput-test test-profile tput
set services rpm rfc2544-benchmarking tests test-name tput-test source-mac-address
00:00:5e:00:53:11
set services rpm rfc2544-benchmarking tests test-name tput-test destination-mac-address
00:00:5e:00:53:22
set services rpm rfc2544-benchmarking tests test-name tput-test ovlan-id 512
set services rpm rfc2544-benchmarking tests test-name tput-test service-type elan
set services rpm rfc2544-benchmarking tests test-name tput-test mode initiate-and-terminate
```

```
set services rpm rfc2544-benchmarking tests test-name tput-test family bridge
set services rpm rfc2544-benchmarking tests test-name tput-test direction egress
set services rpm rfc2544-benchmarking tests test-name tput-test source-udp-port 200
set services rpm rfc2544-benchmarking tests test-name tput-test destination-udp-port 400
set services rpm rfc2544-benchmarking tests test-name tput-test test-iterator-duration 250
set services rpm rfc2544-benchmarking tests test-name tput-test test-interface ge-0/2/0.0
set services rpm rfc2544-benchmarking tests test-name b2bt-test test-profile b2bt
set services rpm rfc2544-benchmarking tests test-name b2bt-test source-mac-address
00:00:5e:00:53:11
set services rpm rfc2544-benchmarking tests test-name b2bt-test destination-mac-address
00:00:5e:00:53:22
set services rpm rfc2544-benchmarking tests test-name b2bt-test ovlan-id 512
set services rpm rfc2544-benchmarking tests test-name b2bt-test service-type elan
set services rpm rfc2544-benchmarking tests test-name b2bt-test mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name b2bt-test family bridge
set services rpm rfc2544-benchmarking tests test-name b2bt-test direction egress
set services rpm rfc2544-benchmarking tests test-name b2bt--test destination-udp-port 400
set services rpm rfc2544-benchmarking tests test-name b2bt-test test-iterator-duration 10
set services rpm rfc2544-benchmarking tests test-name b2b-test test-interface ge-0/2/0.0
set services rpm rfc2544-benchmarking tests test-name lty-test test-profile lty
set services rpm rfc2544-benchmarking tests test-name lty-test source-mac-address
00:00:5e:00:53:11
set services rpm rfc2544-benchmarking tests test-name lty-test destination-mac-address
00:00:5e:00:53:22
set services rpm rfc2544-benchmarking tests test-name lty-test ovlan-id 512
set services rpm rfc2544-benchmarking tests test-name lty-test service-type elan
set services rpm rfc2544-benchmarking tests test-name lty-test mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name lty-test family bridge
set services rpm rfc2544-benchmarking tests test-name lty-test direction egress
set services rpm rfc2544-benchmarking tests test-name lty-test source-udp-port 200
set services rpm rfc2544-benchmarking tests test-name lty-test destination-udp-port 400
set services rpm rfc2544-benchmarking tests test-name lty-test test-iterator-duration 10
set services rpm rfc2544-benchmarking tests test-name lty-test test-interface ge-0/2/0.0
set services rpm rfc2544-benchmarking tests test-name frloss-test test-profile frloss
set services rpm rfc2544-benchmarking tests test-name frloss-test source-mac-address
00:00:5e:00:53:11
set services rpm rfc2544-benchmarking tests test-name frloss-test destination-mac-address
00:00:5e:00:53:22
set services rpm rfc2544-benchmarking tests test-name frloss-test ovlan-id 512
set services rpm rfc2544-benchamrking tests test-name frloss-test ovlan-priority 7
set services rpm rfc2544-benchamrking tests test-name frloss-test ovlan-cfi 1
set services rpm rfc2544-benchmarking tests test-name frloss-test service-type elan
set services rpm rfc2544-benchmarking tests test-name frloss-test mode initiate-and-terminate
```

```
set services rpm rfc2544-benchmarking tests test-name frloss-test family bridge
set services rpm rfc2544-benchmarking tests test-name frloss-test direction egress
set services rpm rfc2544-benchmarking tests test-name frloss-test source-udp-port 200
set services rpm rfc2544-benchmarking tests test-name frloss-test destination-udp-port 400
set services rpm rfc2544-benchmarking tests test-name frloss-test test-iterator-duration 30
set services rpm rfc2544-benchmarking tests test-name frloss-test test-interface ge-0/2/0.0
set interfaces ge-0/2/0 flexible-vlan-tagging
set interfaces ge-0/2/0 mtu 9192
set interfaces ge-0/2/0 encapsulation flexible-ethernet-services
set interfaces ge-0/2/0 unit 0 encapsulation vlan-bridge
set interfaces ge-0/2/0 unit 0 vlan-id 512
set interfaces ge-0/2/1 flexible-vlan-tagging
set interfaces ge-0/2/1 mtu 9192
set interfaces ge-0/2/1 encapsulation flexible-ethernet-services
set interfaces ge-0/2/1 unit 0 encapsulation vlan-bridge
set interfaces ge-0/2/1 unit 0 vlan-id 512
set bridge-domains bd1 vlan-id 10
set bridge-domains bd1 interface ge-0/2/1.0
set bridge-domains bd1 interface ge-0/2/0.0
```

Configuring VPLS Parameters on the MX Router (Provider Edge Router PE1)

```
set chassis fpc 0 pic 2 tunnel-services
set interfaces ge-0/2/1 flexible-vlan-tagging
set interfaces ge-0/2/1 mtu 9192
set interfaces ge-0/2/1 encapsulation vlan-vpls
set interfaces ge-0/2/1 unit 0 encapsulation vlan-vpls
set interfaces ge-0/2/1 unit 0 vlan-id 512
set interfaces ge-0/3/0 mtu 9192
set interfaces ge-0/3/0 unit 0 family inet address 192.0.2.1/24
set interfaces ge-0/3/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 198.51.100.1/32
set routing-options router-id 198.51.100.1
set routing-options autonomous-system 65100
set protocols mpls interface ge-0/3/0.0
set protocols bgp group test type internal
set protocols bgp group test local-address 198.51.100.1
set protocols bgp group test family l2vpn signaling
set protocols bgp group test neighbor 198.51.100.2
set protocols ospf traffic-engineering
set protocols ospf reference-bandwidth 1g
set protocols ospf area 0.0.0.0 interface ge-0/3/0.0
```

```
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ldp interface ge-0/3/0.0 set protocols ldp interface lo0.0
set routing-instances vpls-pe1 instance-type vpls
set routing-instances vpls-pe1 interface ge-0/2/1.0
set routing-instances vpls-pe1 no-local-switching
set routing-instances vpls-pe1 route-distinguisher 198.51.100.1:101
set routing-instances vpls-pe1 vrf-target target:1:2
set routing-instances vpls-pe1 protocols vpls site-range 8
set routing-instances vpls-pe1 protocols vpls no-tunnel-services
set routing-instances vpls-pe1 protocols vpls site HUB site-identifier 1
set routing-instances vpls-pe1 protocols vpls vpls-id 1
set routing-instances vpls-pe1 protocols vpls neighbor 198.51.100.2
```

Configuring Benchmarking Test Parameters and VPLS Parameters on the MX104 Router (Provider Edge Router PE2)

```
set services rpm rfc2544-benchmarking tests test-name l2v-reflector source-mac-address
00:00:5e:00:53:11
set services rpm rfc2544-benchmarking tests test-name l2v-reflector destination-mac-address
00:00:5e:00:53:22
set services rpm rfc2544-benchmarking tests test-name l2v-reflector service-type elan
set services rpm rfc2544-benchmarking tests test-name l2v-reflector in-service
set services rpm rfc2544-benchmarking tests test-name l2v-reflector ip-swap
set services rpm rfc2544-benchmarking tests test-name l2v-reflector udp-tcp-port-swap
set services rpm rfc2544-benchmarking tests test-name l2v-reflector mode reflect
set services rpm rfc2544-benchmarking tests test-name l2v-reflector family vpls
set services rpm rfc2544-benchmarking tests test-name l2v-reflector reflect-etype 2048
set services rpm rfc2544-benchmarking tests test-name l2v-reflector direction egress
set services rpm rfc2544-benchmarking tests test-name l2v-reflector source-udp-port 200
set services rpm rfc2544-benchmarking tests test-name l2v-reflector destination-udp-port 200
set services rpm rfc2544-benchmarking tests test-name l2v-reflector test-interface ge-0/3/1.0
set interfaces ge-0/2/5 mtu 9192
set interfaces ge-0/2/5 unit 0 family inet address 203.0.113.1/24
set interfaces ge-0/2/5 unit 0 family mpls
set interfaces ge-0/3/1 flexible-vlan-tagging
set interfaces ge-0/3/1 mtu 9192
set interfaces ge-0/3/1 encapsulation vlan-vpls
set interfaces ge-0/3/1 unit 0 encapsulation vlan-vpls
set interfaces ge-0/3/1 unit 0 vlan-id 512
set interfaces ge-0/3/1 unit 0 family vpls filter input portmirror
set interfaces ge-0/3/1 unit 0 family vpls filter output portmirror
set interfaces ge-0/3/2 flexible-vlan-tagging
```

```
set interfaces ge-0/3/2 mtu 9192
set interfaces ge-0/3/2 encapsulation vlan-vpls
set interfaces ge-0/3/2 unit 0 encapsulation vlan-vpls
set interfaces ge-0/3/2 unit 0 vlan-id 512
set interfaces lo0 unit 0 family inet address 198.51.100.2/32
set forwarding-options port-mirroring input rate 1
set forwarding-options port-mirroring family vpls output interface ge-0/3/3.0
set forwarding-options port-mirroring family vpls output no-filter-check
set forwarding-options port-mirroring instance pm1 input rate 10000
set forwarding-options port-mirroring instance pm1 family vpls output interface ge-0/3/3.0
set routing-options router-id 198.51.100.2
set routing-options autonomous-system 65100
set protocols mpls interface ge-0/2/5.0
set protocols bgp group test type internal
set protocols bgp group test local-address 198.51.100.2
set protocols bgp group test family l2vpn signaling
set protocols bgp group test neighbor 198.51.100.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/2/5.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ldp interface ge-0/2/5.0
set protocols ldp interface lo0.0
set firewall family vpls filter portmirror term 1 then count pm1
set firewall family vpls filter portmirror term 1 then accept
set firewall family vpls filter portmirror term 1 then port-mirror
set routing-instances vpls-pe2 instance-type vpls
set routing-instances vpls-pe2 interface ge-0/3/1.0
set routing-instances vpls-pe2 interface ge-0/3/3.0
set routing-instances vpls-pe2 no-local-switching
set routing-instances vpls-pe2 route-distinguisher 198.51.100.2:102
set routing-instances vpls-pe2 vrf-target target:1:2
set routing-instances vpls-pe2 protocols vpls site-range 8
set routing-instances vpls-pe2 protocols vpls no-tunnel-services
set routing-instances vpls-pe2 protocols vpls site SPOKE site-identifier 2
set routing-instances vpls-pe2 protocols vpls vpls-id 1
set routing-instances vpls-pe2 protocols vpls neighbor 198.51.100.1
```

Configuring Throughput Benchmarking Test Parameters on the ACX Series Router (Initiator)

Step-by-Step Procedure

The following configuration requires you to configure a test profile for the throughput test and reference the test profile in a unique test name. The test name defines the parameters for the throughput test to be performed on the ACX Series router.

To configure the throughput test parameters on the ACX Series router:

1. In configuration mode, at the [edit] hierarchy level, configure a real-time performance monitoring service (RPM) instance and an RFC2544-based benchmarking test for the RPM instance.

   ```
   [edit]
   user@host# edit services rpm rfc2544-benchmarking
   ```

2. Define a name for the first test profile—for example, tput—for the throughput test profile.

   ```
   [edit services rpm rfc2544-benchmarking]
   user@host# set profiles test-profile tput
   ```

3. Configure the type of test to be performed as throughput, specify the packet size as 256 bytes, and define the theoretical maximum bandwidth for the test as 600000 Kbps. You can specify any value from 1 Kbps through 1,000,000 Kbps for the maximum bandwidth.

   ```
   [edit services rpm rfc2544-benchmarking profiles test-profile tput]
   user@host# set test-type throughput packet-size 256 bandwidth-kbps 600000
   ```

4. Enter the up command twice to go to the [edit services rpm rfc2544-benchmarking] level in the configuration hierarchy.

   ```
   [edit services rpm rfc2544-benchmarking profiles test-profile tput ]
   user@host# up
   user@host# up
   ```

5. Define a name for the throughput test—for example, `tput-test`. The test name can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking ]
user@host# set tests test-name tput-test
```

6. Specify the name of the test profile, `tput`, to be associated with the test name.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set test-profile tput
```

7. Configure the source and destination MAC addresses for the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set source-mac-address 00:00:5e:00:53:11 destination-mac-address
00:00:5e:00:53:22
```

8. Configure the outer VLAN ID for the test frames and specify the service type under test to be E-LAN.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set ovlan-id 512 service-type elan
```

9. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set mode initiate-and-terminate
```

10. Configure the family type, `bridge`, for the benchmarking test and specify the direction, egress. Also, specify the source and destination UDP ports to be used in the UDP headers of the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set family bridge direction egress source-udp-port 200 destination-udp-port 400
```

11. Specify the duration of each iteration in seconds, with a value from 10 seconds to 1,728,000 seconds, and specify the logical interface, ge-0/2/0.0, on which the RFC2544-benchmarking tests are run.

```
[edit services rpm rfc2544-benchmarking tests test-name tput-test]
user@host# set test-iterator-duration 250 test-interface ge-0/2/0.0
```

Configuring Back-to-Back Frames Benchmarking Test Parameters on the ACX Series Router

Step-by-Step Procedure

The following configuration requires you to configure a test profile for the back-to-back frames test and reference the test profile in a unique test name. The test name defines the parameters for the back-to-back frames test to be performed on the ACX Series router.

To configure the back-to-back frames test parameters on the ACX Series router:

1. In configuration mode, at the `[edit]` hierarchy level, configure a real-time performance monitoring service (RPM) instance and an RFC2544-based benchmarking test for the RPM instance.

```
[edit]
user@host# edit services rpm rfc2544-benchmarking
```

2. Define a name for the back-to-back test profile—for example, `b2bt`.

```
[edit services rpm rfc2544-benchmarking]
user@host# set profiles test-profile b2bt
```

3. Configure the type of test to be performed as back-to-back frames, specify the packet size as 9104 bytes, and specify the theoretical maximum bandwidth for the test as 600000 Kbps. You can specify any value from 1 Kbps through 1,000,000 Kbps as the maximum bandwidth.

```
[edit services rpm rfc2544-benchmarking profiles test-profile b2bt]
user@host# set test-type back-toback-frames packet-size 9104 bandwidth-kbps 600000
```

4. Enter the `up` command twice to go to the [`edit services rpm rfc2544-benchmarking`] level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles test-profile b2bt ]
user@host# up
user@host# up
```

5. Define a name for the back-to-back frames test—for example, `b2bt-test`. The test name can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking ]
user@host# set tests test-name b2bt-test
```

6. Specify the name of the test profile, `b2bt`, to be associated with the test name.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set test-profile b2bt
```

7. Configure the source and destination MAC addresses for the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set source-mac-address 00:00:5e:00:53:11 destination-mac-address
00:00:5e:00:53:22
```

8. Configure the outer VLAN ID for the test frames and specify the service type under test as E-LAN.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set ovlan-id 512 service-type elan
```

9. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set mode initiate-and-terminate
```

10. Configure the family type, `bridge`, for the benchmarking test and specify the direction, egress.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host# set family bridge direction egress
```

11. Specify the duration of each iteration in seconds, with a value from 10 seconds to 1,728,000 seconds. Also, specify the logical interface, ge-0/2/0.0, on which the RFC2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name b2bt-test]
user@host#
                        set test-iterator-duration 10 test-interface ge-0/2/0.0
```

Configuring Latency Benchmarking Test Parameters on the ACX Series Router

Step-by-Step Procedure

The following configuration requires you to configure a test profile for the latency test and reference the test-profile in a unique test-name. The test-name defines the parameters for the latency test to be performed on the initiator (ACX Series router).

To configure the latency test parameters on the initiator:

1. In configuration mode, at the `[edit]` hierarchy level, configure a real-time performance monitoring service (RPM) instance and an RFC2544-based benchmarking test for the RPM instance.

```
[edit]
user@host# edit services rpm rfc2544-benchmarking
```

2. Define a name for the latency test profile—for example, `lty`.

```
[edit services rpm rfc2544-benchmarking]
user@host# set profiles test-profile lty
```

3. Configure the type of test to be performed as latency, specify the packet size of the test packet as 1024, and specify the maximum bandwidth for the test in Kbps, with a value from 1 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles]
user@host# set test-profile lty test-type latency packet-size 1024 bandwidth-kbps 600000
```

4. Enter the `up` command twice to go to the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles test-profile lty]
user@host# up
user@host# up
```

5. Define a name for the latency test—for example, `lty-test`. The test name can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking ]
user@host# set tests test-name lty-test
```

6. Specify the name of the test profile, `lty`, to be associated with the test name.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set test-profile lty
```

7. Configure the source and destination MAC addresses for the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set source-mac-address 00:00:5e:00:53:11 destination-mac-address
00:00:5e:00:53:22
```

8. Configure the outer VLAN ID for the test frames and specify the service type under test.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set ovlan-id 512 service-type elan
```

9. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set mode initiate-and-terminate
```

10. Configure the family type, `bridge`, for the benchmarking test and specify the direction, egress. Also, specify the source and destination UDP port to be used in the UDP headers of the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set family bridge direction egress source-udp-port 200 destination-udp-port 400
```

11. Specify the duration of each iteration in seconds, with a value from 10 seconds to 1,728,000 seconds. Also, specify the logical interface, ge-0/2/0.0, on which the RFC2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name lty-test]
user@host# set test-iterator-duration 10 test-interface ge-0/2/0.0
```

Configuring Frame Loss Benchmarking Test Parameters on the ACX Series Router

Step-by-Step Procedure

The following configuration requires you to configure a test profile for the frame loss test and reference the test-profile in a unique test-name. The test-name defines the parameters for the frame loss test to be performed on the ACX Series router.

To configure the frame loss test parameters on the ACX Series router:

1. In configuration mode, at the `[edit]` hierarchy level, configure a real-time performance monitoring service (RPM) instance and an RFC2544-based benchmarking test for the RPM instance.

```
[edit]
user@host# edit services rpm rfc2544-benchmarking
```

2. Define a name for the frame loss test profile—for example, `frloss`.

```
[edit services rpm rfc2544-benchmarking]
user@host# set profiles test-profile frloss
```

3. Configure the type of test performed as frame loss, specify the packet size of the test packet, and define the maximum bandwidth for the test in kilobits per second, with a value from 1 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles]
user@host# set test-profile frloss test-type frame-loss packet-size 1600 bandwidth-kbps
600000
```

4. Enter the `up` command to go to the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles]
user@host# up
```

5. Define a name for the frame loss test—for example, `frloss-test`. The test name can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking ]
user@host# set tests test-name frloss-test
```

6. Specify the name of the test profile, `frloss`, to be associated with the test name.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set test-profile frloss
```

7. Configure the source and destination MAC address for the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set source-mac-address 00:00:5e:00:53:11 destination-mac-address
00:00:5e:00:53:22
```

8. Configure the outer VLAN ID, priority, and the canonical format indicator (cfi) value for the test frames. Together, the four added bytes, priority (3 bits) and canonical format indicator (1 bit) form the VLAN tag. Also, specify the service type under test.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set ovlan-id 512 ovlan-priority 7 ovlan-cfi 1 service-type elan
```

9. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set mode initiate-and-terminate
```

10. Configure the family type, `bridge`, for the benchmarking test and specify the direction, egress. Also, specify the source and destination UDP port to be used in the UDP headers of the test packet.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set family bridge direction egress source-udp-port 200 destination-udp-port 400
```

11. Specify the duration of each iteration in seconds, with a value from 10 seconds to 1,728,000 seconds. Also, specify the logical interface, ge-0/2/1.0, on which the RFC2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name frloss-test]
user@host# set test-iterator-duration 30 test-interface ge-0/2/0.0
```

12. Enter the `exit` command to go to the [edit] hierarchy level.

```
[edit services rpm rfc2544-benchmarking tests test-name test4 ]
user@host# exit
```

Configuring Other Benchmarking Test Parameters on the ACX Series Router

Step-by-Step Procedure

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

To configure the interface and bridge domain on the ACX Series router:

1. Configure the Layer 2 NNI interface on which the tests must be run from the `[edit]` hierarchy level.

```
[edit]
user@host# edit interfaces ge-0/2/1
```

2. Configure flexible VLAN tagging for the transmission of untagged frames or 802.1Q single-tagged and dual-tagged frames on the logical interface. You can also specify the maximum transmission unit (MTU) size for the interface and the encapsulation.

```
[edit interfaces ge-0/2/1]
user@host# set flexible-vlan-tagging mtu 9192 encapsulation flexible-ethernet-services
```

3. Configure a logical unit for the interface, specify the encapsulation, and configure the VLAN ID on the logical interfaces.

```
[edit interfaces ge-0/2/1]
user@host# set unit 0 encapsulation vlan-bridge vlan-id 512
```

4. Configure the Layer 2 UNI interface.

```
[edit]
user@host# edit interfaces ge-0/2/0
```

5. Configure flexible VLAN tagging for transmission of non-tagged frames or 802.1Q single-tag and dual-tag frames on the logical interface. You can also specify the maximum transmission unit (MTU) size for the interface and the encapsulation.

```
[edit interfaces ge-0/2/0]
user@host# set flexible-vlan-tagging mtu 9192 encapsulation flexible-ethernet-services
```

6. Configure a logical unit for the interface and specify the encapsulation and configure the VLAN ID on the logical interfaces.

```
[edit interfaces ge-0/2/0]
user@host# set unit 0 encapsulation vlan-bridge vlan-id 512
```

7. Configure the bridge domain, bd1, and specify the VLAN ID associated with the bridge domain and the associated interfaces from the `[edit]` hierarchy level.

```
[edit]
user@host# set bridge-domains bd1 vlan-id 10 interface ge-0/2/1.0
user@host# set bridge-domains bd1 vlan-id 10 interface ge-0/2/0.0
```

Configuring the VPLS Parameters on the MX Series Router (PE1)

Step-by-Step Procedure

The following configuration requires you to enable a simple VPLS topology between the PE1 and PE2 routers. In this example, PE1 is a MX Series router. On the PE1 router, configure the tunnel services interface and prepare the router for VPLS by configuring the BGP, MPLS, OSPF protocols.

To configure the VPLS parameters on the MX Series router:

1. Configure tunnel services.

```
[edit]
user@host# set chassis fpc 0 pic 2 tunnel-services
```

2. Configure the VPLS VLAN encapsulation on the router.

```
[edit]
user@host# set interfaces ge-0/2/1 flexible-vlan-tagging
user@host# set interfaces ge-0/2/1 mtu 9192
user@host# set interfaces ge-0/2/1 encapsulation vlan-vpls
user@host# set interfaces ge-0/2/1 unit 0 encapsulation vlan-vpls
user@host# set interfaces ge-0/2/1 unit 0 vlan-id 512
```

3. Configure the routing interface and the loopback interface on the router.

```
[edit]
user@host# set interfaces ge-0/3/0 mtu 9192
user@host# set interfaces ge-0/3/0 unit 0 family inet address 192.0.2.1/24
user@host# set interfaces ge-0/3/0 unit 0 family mpls
user@host# set interfaces lo0 unit 0 family inet address 198.51.100.1/32
```

4. Configure the routing options on the router.

```
[edit]
user@host# set routing-options router-id 198.51.100.1
user@host# set routing-options autonomous-system 65100
```

5. Configure MPLS on the router to advertise the Layer 2 VPN interface that communicates with the PE2 router.

```
[edit]
user@host# set protocols mpls interface ge-0/3/0.0
```

6. Configure BGP as the signaling protocol on the router to enable carrying of Layer 2 VPLS messages.

```
[edit]
user@host# set protocols bgp group test type internal
user@host# set protocols bgp group test local-address 198.51.100.1
user@host# set protocols bgp group test family l2vpn signaling
user@host# set protocols bgp group test neighbor 198.51.100.2
```

7. Configure OSPF on the router to enable exchange of routing information.

```
[edit]
user@host# set protocols ospf traffic-engineering
user@host# set protocols ospf reference-bandwidth 1g
user@host# set protocols ospf area 0.0.0.0 interface ge-0/3/0.0
user@host# set protocols ospf area 0.0.0.0 interface lo0.0
```

8. Configure LDP on the router to enable LDP for all connections

```
[edit]
user@host# set protocols ldp interface ge-0/3/0.0
user@host# set protocols ldp interface lo0.0
```

9. Create and configure the VPLS routing interface `vpls-pe1`.

```
[edit]
user@host# set routing-instances vpls-pe1 instance-type vpls
user@host# set routing-instances vpls-pe1 interface ge-0/2/1.0
user@host# set routing-instances vpls-pe1 no-local-switching
user@host# set routing-instances vpls-pe1 route-distinguisher 198.51.100.1:101
user@host# set routing-instances vpls-pe1 vrf-target target:1:2
user@host# set routing-instances vpls-pe1 protocols vpls site-range 8
user@host# set routing-instances vpls-pe1 protocols vpls no-tunnel-services
user@host# set routing-instances vpls-pe1 protocols vpls site HUB site-identifier 1
user@host# set routing-instances vpls-pe1 protocols vpls vpls-id 1
user@host# set routing-instances vpls-pe1 protocols vpls neighbor 198.51.100.2
```

Configuring Benchmarking Test Parameters on the MX104 Router (Reflector)

Step-by-Step Procedure

The following configuration requires you to configure a unique test-name for the benchmarking test on the MX104 Series router. The test-name defines the parameters for the benchmarking test to be performed. Because the test interface and test MAC addresses are the same, you can create a single test configuration at the reflector (MX104).

To configure the benchmarking test parameters on the MX104 Series router:

1. In configuration mode, at the [edit] hierarchy level, configure a real-time performance monitoring service (RPM) instance and an RFC2544-based benchmarking test for the RPM instance.

```
[edit]
user@host# edit services rpm rfc2544-benchmarking
```

2. Define a name for the test—for example, l2v-reflector. The test name can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking ]
user@host# set tests test-name l2v-reflector
```

3. Specify the source and destination MAC addresses of the test packet.

```
[edit services rpm rfc2544-benchmarking test-name l2v-reflector]
user@host# set source-mac-address 00:00:5e:00:53:11 destination-mac-address 00:00:5e:00:53:22
```

4. Specify the service type under test and the mode in which the test is executed, which is in-service, at the reflector. Also, specify if the IP address, TCP and UDP port must be swapped.

```
[edit services rpm rfc2544-benchmarking test-name l2v-reflector]
user@host# set service-type elan in-service ip-swap udp-tcp-port-swap
```

5. Specify the mode, which is reflect, at the reflector.

```
[edit services rpm rfc2544-benchmarking test-name l2v-reflector]
user@host# set mode reflect
```

6. Configure the family type, vpls, specify the direction, egress, and specify the protocol being transported in the Ethernet frame, for the benchmarking test. Also, specify the source and destination UDP ports and specify the logical interface, ge-0/3/1.0, on which the RFC2544-based benchmarking test is being run.

```
[edit services rpm rfc2544-benchmarking tests test-name l2v-reflector]
user@host# set family vpls direction egress source-udp-port 200 destination-udp-port 200 test-
interface ge-0/3/1.0
```

Configuring Other Benchmarking Test Parameters on the MX104 Router (Reflector)

Step-by-Step Procedure

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

To configure the interface and bridge domain on the MX104 Series router:

1. Configure the Layer 2 NNI interface on which the tests must be run.

   ```
   [edit]
   user@host# edit interfaces ge-0/3/1.0
   ```

2. Configure flexible VLAN tagging for transmission of untagged frames or 802.1Q single-tagged and dual-tagged frames on the logical interface. You can also specify the maximum transmission unit (MTU) size for the interface and the encapsulation.

   ```
   [edit interfaces ge-0/3/1.0]
   user@host# set flexible-vlan-tagging mtu 9192 encapsulation vlan-vpls
   ```

3. Configure a logical unit for the interface, specify the encapsulation, and configure the VLAN ID on the logical interface.

   ```
   [edit interfaces ge-0/3/1.0]
   user@host# set unit 0 encapsulation vlan-vpls vlan-id 512
   ```

4. Configure the Layer 2 UNI interface.

   ```
   [edit]
   user@host# edit interfaces ge-0/3/2.0
   ```

5. Configure flexible VLAN tagging for transmission of untagged frames or 802.1Q single-tagged and dual-tagged frames on the logical interface. You can also specify the maximum transmission unit (MTU) size for the interface and the encapsulation.

```
[edit interfaces ge-0/3/2.0]
user@host# set flexible-vlan-tagging mtu 9192 encapsulation vlan-vpls
```

6. Configure a logical unit for the interface, specify the encapsulation, and configure the VLAN ID on the logical interfaces.

```
[edit interfaces ge-0/3/2.0]
user@host# set unit 0 encapsulation vlan-vpls vlan-id 512
```

7. Configure the bridge domain, `bd1`, and specify the VLAN ID associated with the bridge domain, and the associated interfaces from the `[edit]` hierarchy level.

```
[edit]
user@host# set bridge-domains bd1 vlan-id 500 interface ge-1/1/6.0
user@host# set bridge-domains bd1 vlan-id 500 intreface ge-1/1/5.0
```

8. Start the benchmarking test on the reflector.

```
user@host> test services rpm rfc2544-benchamrking test l2v-reflector start
```

After the test is successfully completed at the initiator, you can stop the test at the reflector by entering the `test services rpm rfc2544-benchmarking test l2v-reflector stop` command.

Configuring VPLS Parameters on the MX104 Router (Reflector)

Step-by-Step Procedure

The following configuration requires you to enable a simple VPLS topology between the PE1 and PE2 routers. In this example, PE2 is a MX104 Series router. On the PE2 router, configure the tunnel services interface and prepare the router for VPLS by configuring the BGP, MPLS, OSPF protocols to complement the configuration on PE1.

1. Configure tunnel services.

```
[edit]
user@host# set chassis fpc 0 pic 2 tunnel-services
```

2. Configure the VPLS VLAN encapsulation on the router.

```
[edit]
user@host# set interfaces ge-0/2/5 flexible-vlan-tagging
user@host# set interfaces ge-0/2/5 mtu 9192
user@host# set interfaces ge-0/2/5 encapsulation vlan-vpls
user@host# set interfaces ge-0/2/5 unit 0 encapsulation vlan-vpls
user@host# set interfaces ge-0/2/5 unit 0 vlan-id 512
```

3. Configure the routing interface and the loopback interface on the router.

```
[edit]
user@host# set interfaces ge-0/3/0 mtu 9192
user@host# set interfaces ge-0/3/0 unit 0 family inet address 192.0.2.1/24
user@host# set interfaces ge-0/3/0 unit 0 family mpls
user@host# set interfaces lo0 unit 0 family inet address 198.51.100.1/32
```

4. Configure the routing options on the router.

```
[edit]
user@host# set routing-options router-id 198.51.100.1
user@host# set routing-options autonomous-system 100
```

5. Configure MPLS on the router to advertise the Layer 2 VPN interface that communicates with the PE1 router.

```
[edit]
user@host# set protocols mpls interface ge-0/2/5.0
```

6. Configure BGP as the signaling protocol on the router to enable carrying of Layer 2 VPLS messages.

```
[edit]
user@host# set protocols bgp group test type internal
user@host# set protocols bgp group test local-address 198.51.100.1
user@host# set protocols bgp group test family l2vpn signaling
user@host# set protocols bgp group test neighbor 198.51.100.2
```

7. Configure OSPF on the router to enable exchange of routing information.

```
[edit]
user@host# set protocols ospf traffic-engineering
user@host# set protocols ospf reference-bandwidth 1g
user@host# set protocols ospf area 0.0.0.0 interface ge-0/2/5.0
user@host# set protocols ospf area 0.0.0.0 interface lo0.0
```

8. Configure LDP on the router to enable LDP for all interfaces.

```
[edit]
user@host# set protocols ldp interface ge-0/2/5.0
user@host# set protocols ldp interface lo0.0
```

9. Create and configure the VPLS routing interface, `vpls-pe2`.

```
[edit]
user@host# set routing-instances vpls-pe2 instance-type vpls
user@host# set routing-instances vpls-pe2 interface ge-0/3/1.0
user@host# set routing-instances vpls-pe2 no-local-switching
user@host# set routing-instances vpls-pe2 route-distinguisher 198.51.100.1:101
user@host# set routing-instances vpls-pe2 vrf-target target:1:2
user@host# set routing-instances vpls-pe2 protocols vpls site-range 8
user@host# set routing-instances vpls-pe2 protocols vpls no-tunnel-services
user@host# set routing-instances vpls-pe2 protocols vpls site SPOKE site-identifier 1
user@host# set routing-instances vpls-pe2 protocols vpls vpls-id 1
user@host# set routing-instances vpls-pe2 protocols vpls neighbor 198.51.100.2
```

Results

In configuration mode, confirm your configuration on the ACX Series router, the MX Series router, and the MX104 Series router by entering the show command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

Benchmarking Test Parameters on the ACX Series router:

```
[edit interfaces]
  ge-0/2/0 {
      flexible-vlan-tagging;
      mtu 9192;
      encapsulation flexible-ethernet-services;
      unit 0 {
          encapsulation vlan-bridge;
          vlan-id 512;
      }
 }
 ge-0/2/1 {
      flexible-vlan-tagging;
      mtu 9192;
      encapsulation flexible-ethernet-services;
      unit 0 {
          encapsulation vlan-bridge;
          vlan-id 512;
      }
 }

[edit bridge-domains]
 bd1 {
    vlan-id 600;
    interface ge-0/2/1.0;
    interface ge-0/2/0.0;
 }

[edit services rpm]
  rfc2544-benchmarking {
      profiles {
          test-profile tput {
              test-type throughput
              packet-size 256;
              bandwidth-kbps 600000;
```

```
        }
        test-profile b2bt {
               test-type back-back-frames
               packet-size 9104;
               bandwidth-kbps 600000;
        }
        test-profile lty {
               test-type latency
               packet-size 1024;
               bandwidth-kbps 600000;
         }
         test-profile frloss {
               test-type frameloss
               packet-size 1600;
               bandwidth-kbps 6000000;
         }
         tests {
            test-name tput-test {
                interface ge-0/2/0.0;
                test-profile tput;
                mode initiate-and-terminate;
                source-mac-address 00:00:5e:00:53:11;
                destination-mac-address 00:00:5e:00:53:22;
                ovlan-id 512;
                service-type elan;
                family bridge;
                direction egress;
                source-udp-port 200;
                destination-udp-port 400;
                test-iterator-duration 250;
            }
            test-name b2b-test {
                interface ge-0/2/0.0;
                test-profile b2bt;
                mode initiate-and-terminate;
                source-mac-address 00:00:5e:00:53:11;
                destination-mac-address 00:00:5e:00:53:22;
                ovlan-id 512;
                service-type elan;
                family bridge;
                direction egress;
                destination-udp-port 400;
                test-iterator-duration 10;
```

```
        }
        test-name lty-test {
            interface ge-0/2/0.0;
            test-profile lty;
            mode initiate-and-terminate;
            source-mac-address 00:00:5e:00:53:11;
            destination-mac-address 00:00:5e:00:53:22;
            ovlan-id 512;
            service-type elan;
            family bridge;
            direction egress;
            source-udp-port 200;
            destination-udp-port 400;
            test-iterator-duration 10;
        }
        test-name frloss-test {
            interface ge-0/2/0.0;
            test-profile frloss;
            mode initiate-and-terminate;
            source-mac-address 00:00:5e:00:53:11;
            destination-mac-address 00:00:5e:00:53:22;
            ovlan-id 512;
            service-type elan;
            family bridge;
            direction egress;
            source-udp-port 200;
            destination-udp-port 400;
            test-iterator-duration 30;
        }
    }
}
```

VPLS Parameters on the MX Series router:

```
[edit routing-instances]
vpls-instance vpls-pe1{
  instance-type vpls;
  interface ge-0/2/1.0;
  route-distinguisher 198.51.100.1:101;
  vrf-target target:1:2;
}
[edit]
```

```
protocols {
  vpls {
     vpls-id 1;
     neighbor 198.51.100.2;
     site-range 8;
     no-tunnel-services;
     site HUB {
       site-identifier 1;
     }
  }
}
```

Benchmarking Test Parameters and VPLS Parameters on the MX104 Series router:

```
[edit interfaces]
  ge-0/3/1 {
      flexible-vlan-tagging;
      mtu 9192;
      encapsulation vlan-vpls;
      unit 0 {
          encapsulation vlan-vpls;
          vlan-id 512;
      }
  }

  ge-0/2/5 {
      flexible-vlan-tagging;
      mtu 9192;
      unit 0 {
          family inet address 203.0.113.1/24;
          family mpls;
      }
  }

[edit services rpm]
  rfc2544-benchmarking {
      # Note, When in reflector mode, test profile is not needed
      tests {
          test-name l2v-reflector {
              interface ge-0/3/1.0;
              source-mac-address 00:00:5e:00:53:11;
              destination-mac-address 00:00:5e:00:53:22;
```

```
            mode reflect;
            service-type elan;
            in-service;
            ip-swap;
            udp-tcp-port swap;
            family vpls;
            reflect-etype 2048;
            direction egress;
            source-udp-port 200;
            destination-udp-port 200;
         }
      }
  }

[edit routing-instances]
   vpls-instance vpls-pe2 {
       instance-type vpls;
       interface ge-0/3/1;
       route-distinguisher 198.51.100.2:102;
       vrf-target target:1:2;
   }
[edit]
  protocols {
     vpls {
        vpls-id 1;
        neighbor 198.51.100.1;
        site-range 8;
        no-tunnel-services;
        site SPOKE {
        site-identifier 2;
     }
  }
```

After you have configured the device, enter the `commit` command, in configuration mode.

Verifying the Results of the Benchmarking Test for Layer 2 ELAN Services Using VPLS

IN THIS SECTION

Examine the results of the benchmarking test that is performed on the configured service between the ACX Series router and the MX104 Series router.

Verifying the Benchmarking Test Results

Purpose

Verify that the necessary and desired statistical values are displayed for the benchmarking test that is run on the configured service between the ACX Series router and the MX104 Series router.

Action

In operational mode, enter the `show services rpm rfc2544-benchmarking (aborted-tests | active-tests | completed-tests | summary)` command to display information about the results of each category or state of the RFC 2544-based benchmarking test, such as terminated tests, active tests, and completed tests, for each real-time performance monitoring (RPM) instance.

Meaning

The output displays the details of the benchmarking test that was performed. For more information about the `show services rpm rfc2544-benchmarking` operational command, see `show services rpm rfc2544-benchmarking` topic in the CLI Explorer.

RELATED DOCUMENTATION

Configuring an RFC 2544-Based Benchmarking Test | 806

Example: Configuring RFC2544-Based Benchmarking Tests on an MX104 Router for Layer 2 E-LAN Services in Bridge Domains | 855

Understanding RFC 2544-Based Benchmarking Tests on MX Series Routers and SRX Series Firewalls | 795

Supported RFC 2544-Based Benchmarking Statements on MX Series Routers | 804

CHAPTER 18

Configuring RFC 2544-Based Benchmarking Tests on ACX Series

IN THIS CHAPTER

RFC 2544-Based Benchmarking Tests for ACX Routers Overview

RFC 2544 defines a series of tests that can be used to describe the performance characteristics of network interconnecting devices. RFC2544-based benchmarking test methodology can be applied to a single device under test (DUT), or a network service (set of devices working together to provide end-to-end service). When applied to a service, the RFC2544 test results can characterize the Service-Level-Agreement (SLA) parameters.

RFC 2544 tests are performed by transmitting test packets from a device that functions as the generator or the initiator. These packets are sent to a device that functions as the reflector, which receives and returns the packets back to the initiator.

ACX Series routers support RFC 2544 tests to measure the following:

- Throughput

- Latency
- Frame loss rate
- Back-to-back frames

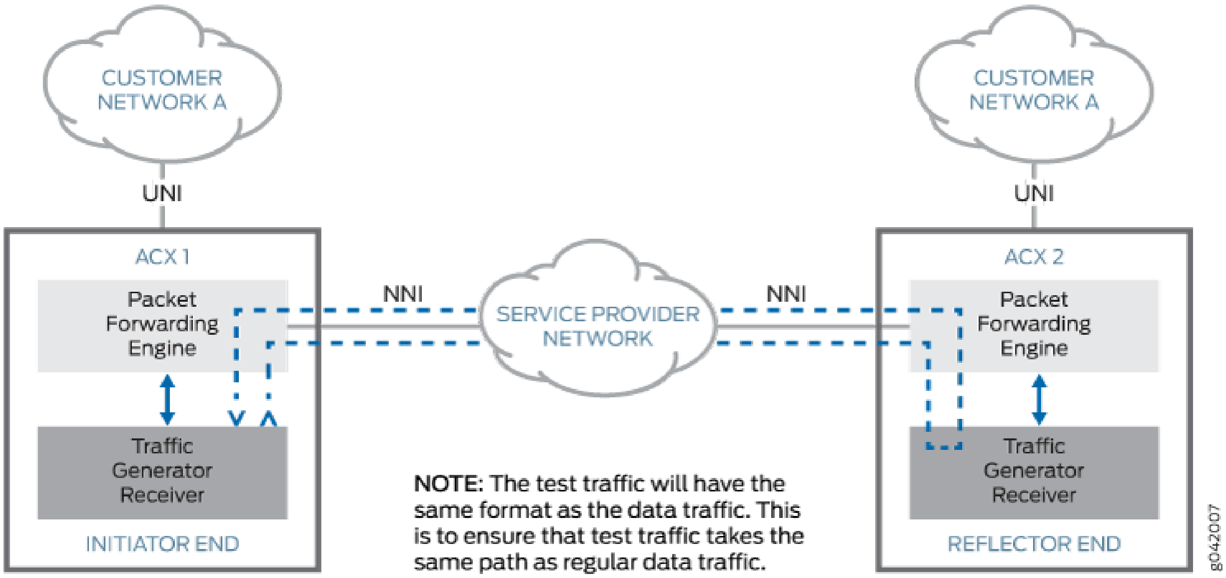
With embedded RFC 2544, an ACX Series router can be configured as an initiator and another ACX Series router as a reflector.

NOTE:

- Prior to Junos OS Evolved 22.4R1, ACX7100 routers can be configured only as a Layer 3 reflector (family `inet`). Starting in Junos OS Evolved 22.4R1, ACX7100 routers can also be configured as a Layer 2 reflector (family `ccc` or `ethernet-switching`.) Starting in Junos OS Evolved Release 24.2R1, these routers can also be configured as Layer 2 ingress reflectors.
- Starting in Junos OS Evolved 22.4R1, ACX7509 and ACX7024 routers can be configured as Layer 2 or Layer 3 reflectors. Starting in Junos OS Evolved Release 24.2R1, these routers can also be configured as Layer 2 ingress reflectors.
- Starting in Junos OS Evolved 23.4R1, ACX7332 and ACX7348 routers can be configured as Layer 2 or Layer 3 reflectors.
- Starting in Junos OS Evolved 23.4R1, ACX7024, ACX7024X, ACX7100, and ACX7509 routers can be configured as Layer 3 initiators.
- Starting in Junos OS Evolved 24.2R1, ACX7024, ACX7024X, ACX7100, ACX7332, and ACX7509 routers can be configured as Layer 2 initiators.
- ACX5448, ACX5048, and ACX5096 routers can be configured only as a Layer 2 reflector (family `bridge` or `ccc`). ACX5048 and ACX5096 routers support only E-Line services.

Figure 73 on page 924 shows the components, role of initiator and reflector, and the flow of test packets in an RFC 2544-based benchmarking test.

Figure 73: RFC 2544-Based Benchmarking Test Methodology



To run RFC 2544-based tests, you need a router to generate service test traffic and a router to reflect the service test traffic back. You need to:

1. Identify two service endpoints between which the RFC2544-based test needs to be run.
2. Configure the reflector end and start reflection.
3. Configure the initiator end and initiate the test.
4. Review the results after the test is complete. Test results are reported in a specific format.

On ACX Series routers, you can run the following RFC 2544-based performance measurement tests:

- Throughput test:
  - Sends a specific number of frames at a specified rate from the initiator through the network service or a DUT. The test starts with a user-configured theoretical maximum rate.
  - Counts the number of transmitted frames and the number of received frames.
    - If the number of frames received is less than those transmitted, the test is repeated with a 50 percent reduced frame rate.
    - Throughput is the maximum rate at which the count of test frames received is equal to the number of test frames transmitted through the network service.

  You can repeat throughput tests for different frame sizes.

- Latency test:

NOTE: To run a latency test, you need to determine the throughput for DUT or a network service at each of the specified frame sizes.

- Starts with a stream of frames at a particular frame size through the DUT at the determined throughput rate.
- Sends an identifying tag in one frame after 60 seconds and calculate the latency when the frame with the same tag is received by the initiator.
- Is repeated for at least 20 times with the reported latency value being the average of the recorded values.

You can repeat latency tests for different frame sizes.

- Frame loss rate test:
  - Involves sending a specific number of frames at a specified rate through the DUT or a network service to be tested and counting the frames that are transmitted.
  - Calculates frame loss rate at each point using the equation:
    ((input_count - output_count) x 100) / input_count.
  - Runs a trial for the frame rate that corresponds to 100 percent of the configured maximum theoretical rate.
  - Is repeated for the frame rate that corresponds to 90 percent of the maximum rate used and then for 80 percent of the maximum rate until a certain trial result shows no lost frames.

  You repeat the frame loss rate tests for different frame sizes.

- Back-to-back frames test:
  - Involves sending a burst of frames with minimum interframe gaps through the DUT or a network service and counting the number of frames forwarded.
  - Is rerun with an increased length of burst frames if the count of transmitted frames is equal to the number of frames forwarded.
  - Is rerun with a reduced length of burst frames if the count of forwarded frames is less than the number of frames transmitted.

  The back-to-back value is the number of frames in the longest burst that the DUT or a network service can handle without the loss of any frames.

  You can repeat back-to-back frame tests for different frame sizes.

Starting in Junos OS Evolved 21.1R1, you can configure RFC 2544-based benchmarking tests on ACX7100 routers. To configure these tests, configure the `rfc2544` statement at the `[edit services monitoring]` hierarchy level.

To configure RFC2544 benchmarking tests for Junos OS, configure the `rfc2544-benchmarking` statement at the `[edit services rpm]` hierarchy level.

The ACX5448 router supports:

- RFC2544 egress Layer 2 reflection functionality for family bridge.
- Multiple RFC2544 reflection sessions.
- Reflection on 1G/10G/40G/Ch10G/Ch25G/100G ports.
- Ethernet Layer 2 frames to carry IP/UDP packets for RFC2544 reflection.

ACX5448 routers do not support the following RFC2544 features:

- Any interface in the bridge domain matching the bridge VLAN identifier.
- Multiple simultaneous sessions with multiple VLAN bridges.
- Multiple test sessions cannot exceed 100G bandwidth.
- IPv6 reflection.
- IPV6 filter support to identify the loopback stream.
- RFC 2544 reflection functionality for family `ccc` (PWE reflection) and family `inet` (Layer 3 IPv4 reflection).
- Reflection without MAC swap and MAC overwrite.
- Reflection on E-Line and E-LAN services.

NOTE: RFC 2544 reflection functionality for family `ccc` (PWE reflection) and family `inet` (Layer 3 IPv4 reflection) is not supported on the ACX710 and ACX5448 routers.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
24.2R1-EVO	Starting in Junos OS Evolved 24.2R1, for ACX7024, ACX7024X, ACX7100, ACX7332, and ACX7509 routers, you can now configure `initiate-and-terminate` mode for Layer 2 test generation (bridge, L2CKT, L2VPN, EVPN-VPWS, EVPN-FXC, EVPN-MPLS, and VPLS) for `family ccc` or `family ethernet-switching` RFC 2544-based benchmarking tests.
24.2R1-EVO	Starting in Junos OS Evolved 24.2R1, for ACX7024, ACX7100-32C, ACX7100-48L, and ACX7509 routers, you can configure RFC 2544-based benchmarking tests with ingress reflection for Layer 2 Ethernet switching services.
23.4R1-EVO	Starting in Junos OS Evolved 23.4R1, we've added support for Layer 2 reflection (bridge, L2CKT, L2VPN, EVPN-VPWS, EVPN-FXC, EVPN-MPLS, and VPLS) with `family ccc` or `family ethernet-switching` and for Layer 3 reflection (IPv4, L3VPN) with `family inet` to the ACX7332 and ACX7348 routers.
23.4R1-EVO	Starting in Junos OS Evolved 23.4R1, for the ACX7024, ACX7024X, ACX7100, and ACX7509 routers, you can now configure `initiate-and-terminate` mode for `family inet` RFC 2544-based benchmarking tests to generate traffic to test Layer 3 services. You can also create test profiles and associate them to particular test names.
22.4R1-EVO	Starting in Junos OS Evolved 22.4R1, we've added support for Layer 2 reflection (bridge, L2CKT, L2VPN, EVPN-VPWS, EVPN-FXC, EVPN-MPLS, and VPLS) with `family ccc` or `family ethernet-switching` to the ACX7100, ACX7509, and ACX7024 routers. We've also added support for Layer 3 reflection (IPv4, L3VPN) with `family inet` to the ACX7509 and ACX7024 routers.
21.1R1-EVO	Starting in Junos OS Evolved 21.1R1, we've added support for Layer 3 reflection (IPv4, L3VPN) with `family inet` for the ACX7100 routers.

RELATED DOCUMENTATION

Layer 2 and Layer 3 RFC 2544-Based Benchmarking Test Overview | 928

Configuring RFC 2544-Based Benchmarking Tests | 933

show services rpm rfc2544-benchmarking

show services rpm rfc2544-benchmarking test-id

Layer 2 and Layer 3 RFC 2544-Based Benchmarking Test Overview

In ACX Series routers, RFC 2544-based benchmark tests can be run to measure the performance characteristics of the E-Line, E-LAN, and EVPL services.

NOTE:

- Prior to Junos OS Evolved Release 22.4R1, ACX7100 routers can be configured only as a Layer 3 reflector (family `inet`). Starting in Junos OS Evolved Release 22.4R1, ACX7100 routers can also be configured as a Layer 2 reflector (family `ccc` or `ethernet-switching`.) Starting in Junos OS Evolved Release 24.2R1, these routers can also be configured as Layer 2 ingress reflectors.

- Starting in Junos OS Evolved Release 22.4R1, ACX7509 and ACX7024 routers can be configured as Layer 2 or Layer 3 reflectors. Starting in Junos OS Evolved Release 24.2R1, these routers can also be configured as Layer 2 ingress reflectors.

- Starting in Junos OS Evolved Release 23.4R1, ACX7332 and ACX7348 routers can be configured as Layer 2 or Layer 3 reflectors.

- Starting in Junos OS Evolved Release 23.4R1, ACX7024, ACX7024X, ACX7100, and ACX7509 routers can be configured as Layer 3 initiators.

- Starting in Junos OS Evolved 24.2R1, ACX7024, ACX7024X, ACX7100, ACX7332, and ACX7509 routers can be configured as Layer 2 initiators.

- ACX5448, ACX5048, and ACX5096 routers can be configured only as a Layer 2 reflector (family `bridge` or `ccc`). ACX5048 and ACX5096 routers support only E-Line services.

- You can configure the test on the following underlying services:

  - Between two IPv4 endpoints—In this mode, the generator sends test packets to a user-configured IP destination or UDP port (which is on the reflector).

  - Between two user-to-network interfaces (UNIs) of Ethernet Virtual Connection (EVC), Ethernet Private Line (EPL, also called E-Line), Ethernet Virtual Private Line (EVPL), EVC (EPL, EVPL)—One end is configured as the generator or initiator and the other end acts as the reflector. The generator receives the test packets that are returned from the reflector and computes the test results.

  NOTE: Benchmarking tests are not supported for IPv6-based services.

- You cannot perform multiple simultaneous RFC 2544-based benchmarking tests on the same pseudowire.
- Interoperation of the RFC 2544 benchmarking tests with other third-party customer premises equipment (CPE) that provides embedded or dedicated benchmarking test capability is not supported.
- Fragmented test-frames and one-way measurements of frames are not supported. You must configure one end or the source device to initiate and terminate test frames and the other end or the destination device to reflect the received frames back to the initiator.
- RFC 2544 generator and reflector are supported with testing bandwidth up to 1 Gbps. ACX5048 and ACX5096 routers supports test bandwidth of up to 40 Gbps.
- RFC2544 Layer 2 reflection supports these Layer 2 services : L2 (Bridge), L2CKT, L2VPN, EVPN-VPWS, EVPN-FXC, EVPN-MPLS, and VPLS. You can configure Layer 2 reflection only in the egress direction. Starting in Junos OS Evolved Release 24.2R1, the ACX7024, ACX7100, and ACX7509 routers can also be configured as Layer 2 ingress reflectors. Layer 2 reflection occurs at the UNI interface for unicast traffic only. The MAC addresses are always swapped after reflection. You can configure swapping for IP addresses or UDP ports using the `ip-swap` or `udp-tcp-port-swap` statements.
- RFC2544 Layer 3 reflection supports IPv4 or L3VPN traffic. You can configure Layer 3 reflection only in the ingress direction. Only traffic destined for the host is reflected; transit traffic is not affected. The IP addresses and UDP addresses are swapped after reflection.
- Supported transport mechanisms for ELAN include:
  - Multipoint Q-in-Q over provider bridged networks
  - Provider Backbone Bridge (Mac-in-Mac)
  - VPLS over IP/MPLS
  - Ethernet VPN, EVPN-MPLS, and EVPN-VXLAN
- Supported transport mechanisms for E-Line include:
  - Ethernet pseudo-wires
  - Q-in-Q
  - Provider Backbone Bridge (Mac-in-Mac)
  - Bridge domains with two logical interfaces; one for service provider and one for customer VLANs.
- The test session is supported in out-of-service mode for the underlying service. You must not transmit any traffic to the UNI port, configured as a generator or a reflector, that is being tested

during the duration of the test. However, other services that are not configured for the testing session are not impacted.

- Devices embedded with benchmarking test capabilities (generators and reflectors) interoperate with other Juniper Networks devices that support the RFC 2544-based generator or reflector functionality.
- RFC 2544 generator traffic undergoes the same traffic classifier and policer or shaper processing as the ingress customer traffic from the UNI port.
- RFC 2544 generator produces a report with clear details of pass or fail for each critical testing metric, based on the configured thresholds.
- The testing packets can be configured and the format of the packet depends on the underlying service on which the test is configured. For IP-based service, the IP or port values can be configured. For Ethernet-based service, unicast untagged or VLAN ID-tagged dot1p formats (IEEE 802.1p or packet classification Layer 2 headers) are supported. The Ethernet destination address and source address that you configured are used.
- You can run RFC 2544 benchmarking `inet` tests on Layer 3 VPNs or virtual routers.
- For an `inet` service, each test session needs to use a unique UDP port. On the initiator device, the source UDP port that you specify by using the `source-udp-port` statement must be unique and not used by other UDP services that terminate at the initiator. On the reflector device, the UDP port of the destination to be used in the UDP header for the generated frames by using the `destination-udp-port` statement must be unique and not used by other UDP services that terminate at the reflector.
- You must start the test on the router that operates as the reflector before you start the test on router that functions as the initiator.
- You must configure the size of the test packet based on the configured MTU of the packets.
- For computation of the test results for a user-to-network interface (UNI) or ingress direction of an Ethernet pseudowire service, the customer edge (CE) device that is configured as a reflector for `inet` must have the reflected destination address resolved using ARP or a statically configured route must be present on the CE device to connect to the initiator.
- For benchmarking tests on the UNI direction of an Ethernet pseudowire service, if reflection mode is configured, you must configure a static ARP entry. Otherwise, the tests fail when test frames on the UNI interface are reflected. ARP resolution does not enable a successful reflection of test frames for UNI interfaces.
- For a CCC family and with the test performed in the egress or network-to-network interface (NNI) direction, the tests stop on the initiator and reflector when the pseudowire goes down.
- For an RFC 2544 test that is run in the egress or network-to-network interface (NNI) direction of an Ethernet service for a CCC family, the ingress features are not applied.

- In ACX5048 and ACX5096 routers, for a CCC family, the pseudowire has to be opened prior to the start of the RFC 2544 test and during the course of the test.

- The configured packet size denotes the untagged packet size. Any additional VLAN in the payload causes the packet length to be increased correspondingly.

- For an `inet` service, if you configure an interface on an initiator for the RFC 2544-based benchmarking test to be run without specifying the source IPv4 address for the test frames, the primary IP address of the interface is used for the test frames. If the primary IP address is not configured, the first IPv4 address of the interface is used. Similarly, for an unnumbered interface on an initiator on which the RFC 2544 test is run, the primary or the first IP address of the donor loopback interface is retrieved and used in the test frames. You must explicitly configure the source IPv4 address for the test frames by using the `source-ipv4-address` statement if you want a particular address to be used.

- RFC 2544 test generates packets for performance benchmarking testing. The packets can be destined for known or unknown unicast MAC addresses, and they can be either tagged or untagged frames. UDP/IP packet is used as the frame payload. Refer to "Configuring RFC 2544-Based Benchmarking Tests" on page 933 for the frame fields that can be configured.

- Supported outer TPIDs for tagged frames are 0x8100, 0x88a8, 0x9100, and 0x9200.

- RFC 2544 benchmark tests can be run in **out-of-service** and in **in-service** modes.

> **NOTE**: In **out-of-service** mode, while the test is running, all the data traffic sent to and from the UNI port under test on the service is interrupted. Control protocol packets are not interrupted.
>
> In **in-service** mode, while the test is running, only the data traffic corresponding to the test session is interrupted, rest of the data traffic flow sent to and from the UNI port under test on the service are not affected. Control protocol packets are not interrupted.

- The source MAC address, destination MAC address, and the UNI port under test configured uniquely identifies the RFC 2544 benchmark test session (or test stream).

- You can run only one test at a time. Multiple simultaneous tests cannot be run at a time.

- The maximum theoretical test bandwidth supported by ACX Series routers for RFC 2544 test initiator or reflector is 1 Gbps. On ACX5048 and ACX5096 routers, the maximum theoretical test bandwidth supported for RFC 2544 reflector is 40 Gbps.

- RFC 2544 tests can be run with different frame sizes. In ACX Series routers, the supported frame sizes are 64, 68, 72, 128, 256, 512, 768, 1024, 1280, 1518, 1522, 1600, 1728, 2496, 3584, 4016, 9104, and 9136 bytes.

- The test uses round-trip traffic for performance measurement.

- A history of the test results is stored in memory.
- The test results can be copied to the local file system or a remote file system, optionally.

> **NOTE**: RFC 2544 tests cannot compute the performance attributes of multicast or broadcast traffic streams.

For ACX routers running Junos OS Evolved, the number of RFC 2544 reflector test sessions supported varies according to the interface speed as shown in Table 138 on page 932.

Table 138: Number of RFC2544 Reflector Sessions Supported

Interface Speed	ACX7509	ACX7100	ACX7024
10G	16 sessions	16 sessions	4 sessions
25G	12 sessions	16 sessions	4 sessions
50G	6 sessions	16 sessions	Not supported
100G	3 sessions	16 sessions	1 session
400G	None	3 sessions	None

For ACX routers running Junos OS Evolved that are initiating Layer 3 tests, we support 64 sessions per Packet Forwarding Engine.

RELATED DOCUMENTATION

RFC 2544-Based Benchmarking Tests for ACX Routers Overview | 922
Configuring RFC 2544-Based Benchmarking Tests | 933
show services rpm rfc2544-benchmarking
show services rpm rfc2544-benchmarking test-id

Configuring RFC 2544-Based Benchmarking Tests

IN THIS SECTION

This topic describes how to configure a `test-profile` and a `test-name`, start and stop a RFC2544-benchmark test, and copy the test result to a local or a remote file.

Test Profile and Test Name Overview

To configure a RFC 2544 benchmark test on an initiator, you must first configure a `test-profile` and reference the `test-profile` in a unique `test-name`. The `test-name` defines the parameters for the tests to be performed.

To configure a `test-profile`, include the `test-profile` *`profile-name`* statement at the `[edit services rpm rfc2544-benchmarking]` hierarchy level. Test profile is applicable only for the initiator, not the reflector.

To configure a `test-name`, include the `test-name` *`test-name`* statement at the `[edit services rpm rfc2544-benchmarking]` (Junos OS) or [edit services monitoring rfc2544 tests] (Junos OS Evolved) hierarchy level.

(Junos OS) To configure Ethernet loopback as the test mode on a logical interface, include the `Ethernet-loopback` statement at the `[edit services rpm rfc2544-benchmarking]` hierarchy level.

The table below lists the parameters for configuring the test profile at the initiator.

Table 139: Parameters for test-profile Configuration at the Initiator

Parameters	Description
`test-type`	RFC 2544 test type (`throughput | latency | frame-loss | back-back-frames`).

Table 139: Parameters for test-profile Configuration at the Initiator *(Continued)*

Parameters	Description
`packet-size`	Size of the test packet. The valid packet sizes are 64, 68, 72, 128, 256, 512, 768, 1024, 1280, 1518, 1522, 1600, 1728, 2496, 3584, 4016, 9104, and 9136 bytes.
`bandwidth-kbps`	Define the maximum bandwidth limit, in kilobits per second (kbps). Range: 1,000 kpbs through 1,000,000 kbps.
`step-percent`	Specify the step percentage for frame-loss tests. Default: 10 percent Range: 1 through 100 percent

The table below lists the parameters for configuring a test-name at initiator and reflector.

Table 140: Parameters for test-name Configuration at the Initiator and the Reflector

Parameters	Description
`check-test-interface-mtu`	When the `check-test-interface-mtu` parameter is configured, the software validates the MTU size of the test packets with the MTU size configured on the interface and the following would be the behavior for initiator and reflector modes: • On the initiator, if the MTU size of the test packet is larger than the MTU size configured on the interface, then the RFC2544-based benchmarking test fails to start. • On the reflector, if the test packets coming to the reflector does not confirm to the MTU size configured on the interface, then these test packets do not get reflected and are dropped.
`destination-ipv4-address`	Specify the destination IPv4 address. This parameter is mandatory when family `inet` is specified and optional when family `ccc` is specified. If a value is not specified, then by default 192.168.1.20 is used. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.

Table 140: Parameters for test-name Configuration at the Initiator and the Reflector *(Continued)*

Parameters	Description
`destination-mac-address`	Specify the destination MAC address. For example, 0011.2233.4455. This parameter cannot be used when family `inet` is specified. This parameter is optional when family `ccc` and `ethernet-switching` is specified. If not specified, then the default value of 0x00:0x11:0xAE:0x92:0x2F:0x28 is used.
`destination-udp-port`	Specify the destination UDP port number for the test frames. Default: 4041. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
`direction`	Specify the test direction (`egress | ingress`). This parameter is valid only when family `ccc`, `ethernet-switching` and `bridge`. This parameter is mandatory for mode `ethernet-loopback`
`disable-signature-check`	Disable signature verification on the received test frames.
`dscp-code-points`	Specify the value of the Differentiated Services (DiffServ) field. For example, 001111. If a value is not specified, then '0' is used in IP header. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
`family`	Configure the test family (`bridge | ccc | ethernet-switching | inet`). This parameter is mandatory for mode `ethernet-loopback`
`forwarding-class`	Specify the forwarding class to be used for test frames.
`halt-on-prefix-down`	If specified, a prefix that moves to the down state causes the corresponding tests to be stopped. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.

Table 140: Parameters for test-name Configuration at the Initiator and the Reflector *(Continued)*

Parameters	Description
ignore-test-interface-state	When the ignore-test-interface-state parameter is configured for RFC2544 benchmarking tests, the test continues to run even if there are any occurrences of interface up or down events. This is applicable to both initiator and reflector test modes.
in-service	If specified, only the data traffic corresponding to the test session is interrupted, rest of the data traffic flow sent to and from the UNI port under test on the service are not affected. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
ivlan-cfi	CFI bit used in the inner VLAN tag. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
ivlan-id	Configure inner VLAN ID for the test frames. This parameter is valid only for family ccc mode. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
ivlan-priority	Configure the priority value for the IEEE 802.1p bit in the inner VLAN tag. Range: 0 through 7. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
mode	Specify the test mode (ethernet-loopback, initiate-and-terminate, or reflect). • ethernet-loopback—Test frames are loopbacked to the measuring device after the source MAC address and the destination MAC addresses are swapped. • initiate-and-terminate—Test frames are initiated and terminated at the same end. If you specify this mode, then a reflector should be configured on the peer end to bring back the test frames. • reflect—Test frames are reflected on the chosen service.

Table 140: Parameters for test-name Configuration at the Initiator and the Reflector *(Continued)*

Parameters	Description
`outer-tag-protocol-id`	TPID to be used in the outer VLAN tag. Supported values are 0x8100, 0x88a8, 0x9100, 0x9200. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
`ovlan-cfi`	CFI bit used in the outer VLAN tag. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
`ovlan-id`	Configure the outer VLAN ID for the test frames. Range: 0 through 4094 This parameter is valid only for family `ccc` mode.
`ovlan-priority`	Configure the priority value for the IEEE 802.1p bit in the outer VLAN tag. Range: 0 through 7 **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
`packet-loss-priority`	Specify the packet loss priority (PLP) value. If a value is not configured, then the default value of `low` is used. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
`reflect-etype`	Specify the EtherType ID to be used for reflection of test frames. This parameter is valid only in mode `reflect`. If not specified, then all EtherTypes are reflected. Range: 1 through 65,535. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.

Table 140: Parameters for test-name Configuration at the Initiator and the Reflector *(Continued)*

Parameters	Description
`reflect-mode`	Specify the reflection mode (`mac-rewrite` | `mac-swap` | `no-mac-swap`). • `mac-rewrite`—MAC values specified in `source-mac-address` and `destination-mac-address` would be used. • `mac-swap`—Swaps the source-mac-adddress and destination-mac-address in the test frame. This is the default behavior. • `no-mac-swap`—Does not swap MAC addresses. Test frames are returned back as-is.
`reflector-port`	Port used to configure reflector functionality for RFC 2544 test. The range of ports that can be used based on the front panel port number are: • On ACX5048 [16 through 53] • On ACX5096 [64 through 95, 100 through 103].
`service-type`	Specify the service type (E-Line or E-LAN)
`skip-arp-iteration`	This parameter is valid only in family `inet` mode. ARP iteration is a 3-second iteration that is run for all `inet` tests. The results of ARP iteration are ignored in test result calculations. The primary use of sending test frames for 3 seconds is to ensure that all devices on the path to destination build their ARP entries. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
`source-ipv4-address`	Specify the source IPv4 address used for the test frames. If a value is not specified for this parameter, then: • For family `ccc`, if a value is not specified, then by default 192.168.1.10 is used. • For family `inet`, the source address of the interface is used to send out test frames. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.

Table 140: Parameters for test-name Configuration at the Initiator and the Reflector ***(Continued)***

Parameters	Description
source-mac-address	Specify the source MAC address. For example, 0011.2233.4455 This parameter cannot be used when family inet is specified. This parameter is optional when family ccc or ethernet-switching is specified. If not specified, then the default value of 0x00:0x60:0x67:0x71:0xC6:0x62 is used.
source-udp-port	Specify the source UDP port number for the test frames. Default: 4040 **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
test-finish-wait-duration	Number of seconds to wait after transmitting the last frame and before concluding that the test as complete. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
test-iterator-duration	Specify the duration of each iteration in seconds. Range: 10 through 120 seconds The default value for test types throughput, back-to-back frames and frame loss rate is 20 seconds. The default value for test type latency is 120 seconds. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
test-interface	Specify the name of the logical interface (UNI) on which the test needs to be run. When you specify the family as inet and mode as initiate-and-terminate the test-interface is ignored. Instead, the test is run on the egress logical interface that is determined by the route lookup on the specified destination-ipv4-address. When you specify the family as inet and mode as reflect, the test-interface is used as the interface to enable reflection service. If test-interface is not configured, a lookup is performed on the source-ipv4-address parameter to determine the interface hosting the address. This parameter is mandatory for mode ethernet-loopback.

Table 140: Parameters for test-name Configuration at the Initiator and the Reflector *(Continued)*

Parameters	Description
`test-profile`	Specify the name of the `test-profile` to be used for the test. The `test-profile` parameter is ignored when mode `reflect` is used. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
`vlan-cfi`	CFI bit used in the VLAN tag. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
`vlan-id`	Configure the VLAN ID for the test frames. This parameter is valid only for mode `ethernet-loopback`. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.
`vlan-priority`	Configure the VLAN priority value. Range: 0 through 7. **NOTE**: This parameter is not supported on ACX5048 and ACX5096 routers.

Configure a Test Profile for an RFC 2544-Based Benchmarking Test

You can configure a test profile by including the `test-profile` *`profile-name`* statement at the `[edit services rpm rfc2544-benchmarking]` hierarchy level.

This procedure shows how to configure a test profile for Junos OS. Routers running Junos OS Evolved only support reflector mode, and so you cannot configure a test profile on these routers.

To configure a test profile:

1. In configuration mode, go to the `[edit services]` hierarchy level.

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

2. Configure an RPM service instance.

```
[edit services]
user@host# edit rpm
```

3. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@host# edit rfc2544-benchmarking
```

4. Define a name for a test profile—for example, profile1.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit profiles test-profile profile1
```

5. Define the theoretical maximum bandwidth for the test in kilobits per second, with a value from 1,000 Kbps through 1,000,000 Kbps. Specify a complete decimal number.

```
[edit services rpm rfc2544-benchmarking profiles test-profile profile1]
user@host# set bandwidth-kbps kbps
```

6. Specify the size of the test packet in bytes, with a value from 64 through 9136, to be used for each test iteration. You can specify up to 10 packet sizes, separated by a space, that are used sequentially for the test. The valid packet sizes are 64, 68, 72, 128, 256, 512, 768, 1024, 1280, 1518, 1522, 1600, 1728, 2496, 3584, 4016, 9104, and 9136 bytes. If you specify a packet size other than the ones listed here as valid sizes, the configuration is saved when you commit the setting and no error message is displayed. However, when you start the test by entering the `test services rpm rfc2544-benchmarking test test-name start` command, an error message is displayed specifying that you configured an invalid packet size in the test profile associated with the test name.

NOTE:

- The minimum frame size for untagged frames should be 64.
- The minimum frame size for single-tagged frames should be 68.
- The minimum frame size for dual-tagged frames should be 72.

These values are no applicable for `inet`.

```
[edit services rpm rfc2544-benchmarking profiles test-profile profile1]
user@host# set packet-size bytes
```

7. Specify the step percentage for frame-loss tests with a value from 1 through 100. This parameter is not applicable for other test types.

```
[edit services rpm rfc2544-benchmarking profiles test-profile profile1]
user@host# set step-percent percent-value
```

8. Configure the type of test to be performed.

   - To configure a throughput test, use the `throughput` option with the `test-type` statement.

   ```
   [edit services rpm rfc2544-benchmarking profiles test-profile profile1]
   user@host# set test-type throughput
   ```

   - To configure a latency test, use the `latency` option with the `test-type` statement.

   ```
   [edit services rpm rfc2544-benchmarking profiles test-profile profile1]
   user@host# set test-type latency
   ```

   - To configure a frame-loss test, use the `frame-loss` option with the `test-type` statement.

   ```
   [edit services rpm rfc2544-benchmarking profiles test-profile profile1]
   user@host# set test-type frame-loss
   ```

   - To configure a back-to-back frames test, use the `back-back-frames` option with the `test-type` statement.

   ```
   [edit services rpm rfc2544-benchmarking profiles test-profile profile1]
   user@host# set test-type back-back-frames
   ```

Configure a Test Name for an RFC 2544-Based Benchmarking Test For an Initiator

You can configure a test name by including the `test-name` *`test-name`* statement at the `[edit services rpm rfc2544-benchmarking]` (Junos OS) or [edit services monitoring rfc2544] (Junos OS Evolved) hierarchy level.

Routers running Junos OS Evolved support only reflector mode.

(Junos OS) To configure a test name and define its attributes for initiator mode:

1. Navigate to the correct hierarchy level in configuration mode.

```
[edit]
user@host# edit services rpm rfc2544-benchmarking
```

2. Define a name for the test—for example, `test1`.

   The test name identifier can be up to 32 characters in length. This step sets the correct hierarchy level for the rest of the steps in this procedure.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

3. Configure the destination IPv4 address for the test packets.

   This parameter is required only if you configure an IPv4 family `inet`. If you do not configure the destination IPv4 address, the default value of 192.168.1.20 is used.

```
user@host# set destination-ipv4-address address
```

4. (Optional) Specify the source MAC address used in generated test frames.

   You configure this statement for family `ccc`; you cannot configure it for an `inet` family. If you specify this parameter for an `inet` family, a commit error occurs when you commit the configuration. If you do not configure the source MAC address, the default value of 0x00:0x60:0x67:0x71:0xC6:0x62 is used.

```
user@host# set source-mac-address address
```

5. Specify the destination MAC address used in generated test frames.

```
user@host# set destination-mac-address address
```

6. Specify the logical interface on which the RFC 2544-based benchmarking test is run.

   This interface is a local user-to-network interface (UNI) on behalf of which the test frames are generated when the test direction is egress.

```
user@host# set test-interface interface-name
```

7. Specify the family for the benchmarking test.

   The `inet` option indicates that the test is run on an IPv4 service. The `ccc` option indicates that the test is run on an CCC or Ethernet pseudowire service. The `bridge` option indicates that the test is run on a Layer 2 service.

   ```
   user@host# set family bridge
   ```

8. Specify the `initiate-and-terminate` mode for the packets that are sent during the benchmarking test.

   The `initiate-and-terminate` option causes the test frames to be initiated from one end and terminated at the same end. The initiation and termination mode requires a reflector to be configured at the peer end to return the test frames from the peer to the originator.

   ```
   user@host# set mode initiate-and-terminate
   ```

9. Specify the direction (`egress | ingress`) of the interface on which the test must be run.

   The `egress` option causes the test to be run in the egress direction of the interface (traffic sent from user-to-network interface (UNI) toward network-to-network interface (NNI)). The `ingress` option causes the test to be run in the ingress direction of the interface (traffic sent on user-to-network interface (UNI)). You cannot configure `ingress` for a `bridge` family.

   ```
   user@host# set direction egress
   ```

10. Configure the outer VLAN ID for the test frames.

    This statement is valid only for a CCC or an Ethernet pseudowire family.

    ```
    user@host# set ovlan-id number
    ```

11. Configure the inner VLAN ID for the test frames.

    This statement is valid only for a CCC or an Ethernet pseudowire family.

    ```
    user@host# set ivlan-id number
    ```

12. Configure the priority value for the IEEE 802.1p bit in the outer VLAN tag.

    The priority value is configured when the UNI interface is dual-tagged.

    ```
    user@host# set ovlan-priority value
    ```

13. (Optional) Configure the priority value for the IEEE 802.1p bit in the inner VLAN tag.

```
user@host# set ivlan-priority value
```

14. (Optional) Configure the CFI value for the outer VLAN tag.

```
user@host# set ovlan-cfi value
```

15. (Optional) Specify the source IPv4 address to be used in generated test frames.

    If you do not configure the `source-ipv4-address` for an `inet` family, the source address of the interface is used to transmit the test frames. If you do not configure the `source-ipv4-address` for a `ccc` family, the default value of 192.168.1.10 is used.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set source-ipv4-address address
```

16. Specify the destination IPv4 address to be used in generated test frames.

```
user@host# set destination-ipv4-address address
```

17. Specify the source UDP port to be used in the UDP header for the generated frames.

    If you do not specify the UDP port, the default value of 4040 is used.

```
user@host# set source-udp-port port-number
```

18. Specify the destination UDP port to be used in the UDP header for the generated frames.

    If you do not specify the UDP port, the default value of 4041 is used.

```
user@host# set destination-udp-port port-number
```

19. Specify the value of the Differentiated Services (DiffServ) field within the IP header of the test frames.

    The DiffServ code point (DSCP) bits value must be set to a valid 6-bit pattern. If you do not specify this value, 0 is used in the DSCP fields in the IP header.

```
user@host# set dscp-code-points dscp-code-bits
```

20. Specify the forwarding class to be used for test frames. The forwarding class specifies the manner in which the test frames are processed by the Packet Forwarding Engine of the router. If you do not configure this parameter, test frames are treated as best-effort traffic.

    ```
    user@host# set forwarding-class class-name
    ```

21. Specify the `halt-on-prefix-down` option to enable a prefix that moves to the down state to cause the corresponding tests to be stopped.

    The `show` command output for the test displays that the test was terminated because the prefix went down. By default, the RFC 2544-based benchmarking test ignores a prefix-down event (when the prefix associated with the test goes down) and continues to run.

    ```
    user@host# set halt-on-prefix-down
    ```

22. Specify the duration of each iteration in seconds.

    If you configure this value, the default value of each iteration depends on the type of test being run. For throughput, back-back-frames, and frame-loss types of tests, the default value is 20 seconds. For latency tests, the default value is 120 seconds.

    ```
    user@host# set test-iterator-duration seconds
    ```

23. Specify the name of the test profile to be associated with a particular test name.

    You must have previously configured the profile by using the `test-profile` *`profile1`* statement at the `[edit services rpm rfc2544-benchmarking]` hierarchy level. The test profile is required when the test mode is configured as initiation and termination. The `test-profile` *`profile1`* parameter is disregarded when the test mode is configured as reflection. A reflection service does not use the parameters specified in the test profile because the reflection service uses the same parameters for the test frames as the received test frames when it returns the frames to the initiator.

    ```
    user@host# set test-profile profile1
    ```

Configure a Test Name for an RFC 2544-Based Benchmarking Test For a Reflector

To configure a test name and define its attributes for reflector mode:

> **NOTE**: In ACX5048 and ACX5096 routers, while performing a RFC 2544 benchmark test, you must ensure that there are no configurations associated with the reflector port.

1. Navigate to the correct hierarchy level in configuration mode:

a. For Junos OS:

```
[edit]
user@host# edit services rpm rfc-benchmarking
```

b. For Junos OS Evolved:

```
[edit]
user@host# edit services monitoring rfc2544
```

2. Define a name for the test—for example, `test1`.

   The test name identifier can be up to 32 characters in length. This step sets the correct hierarchy level for the rest of the steps in this procedure.

   a. For Junos OS:

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

   b. For Junos OS Evolved:

```
[edit services monitoring rfc2544]
user@host# edit tests test-name test1
```

3. Specify the test mode for the packets that are sent during the benchmarking test.

   The `reflect` option causes the test frames to be reflected back to the initiator end.

```
user@host# set mode reflect
```

4. Specify the family for the benchmarking test.

   Configure the `bridge` option for Junos OS or the `ethernet-switching` option for Junos OS Evolved.

   a. For Junos OS:

```
user@host# set family bridge
```

b. For Junos OS Evolved:

```
user@host# set family ethernet-switching
```

5. Specify the direction (`egress | ingress`) of the interface on which the test must be run.

   The `egress` option causes the test to be run in the egress direction of the interface (traffic sent from user-to-network interface (UNI) toward network-to-network interface (NNI)). The `ingress` option causes the test to be run in the ingress direction of the interface (traffic sent on user-to-network interface (UNI)). You cannot configure `ingress` for a `bridge` or `ethernet-switching` family.

   ```
   user@host# set direction egress
   ```

6. Configure the destination IPv4 address for the test packets.

   You configure this statement only if you configure the IPv4 family `inet` option. This option is not required if you specify circuit cross-connect (CCC) or `ethernet-switching` as the family. If you do not configure the destination IPv4 address, the default value of 192.168.1.20 is used.

   ```
   user@host# set destination-ipv4-address address
   ```

7. Specify the source MAC address used in generated test frames.

   You configure this statement for a `ccc` or `ethernet-switching` family and not for an `inet` family. If you specify this parameter for an `inet` family, a commit error occurs when you commit the configuration. This parameter is optional. If you do not configure the source MAC address, the default value of 0x00:0x60:0x67:0x71:0xC6:0x62 is used.

   ```
   [edit services rpm rfc2544-benchmarking tests test-name test1]
   user@host# set source-mac-address address
   ```

8. Specify the destination MAC address used in generated test frames.

   ```
   user@host# set destination-mac-address address
   ```

9. Specify the logical interface on which the RFC 2544-based benchmarking test is run.

   This interface is a local user-to-network interface (UNI) on behalf of which the test frames are generated when the test direction is egress. You cannot configure this statement for Layer 3 reflection (family `inet`).

   ```
   user@host# set test-interface interface-name
   ```

10. Specify the service type as E-Line or E-LAN.

```
user@host# set service-type eline | elan
```

11. (Junos OS) Specify the forwarding class to be used for test frames.

    The forwarding class specifies the manner in which the test frames are processed by the Packet Forwarding Engine of the router. If you do not configure this parameter, test frames are treated as best-effort traffic.

```
user@host# set forwarding-class class-name
```

12. (Optional) Specify the EtherType to be used for reflection of the test frames.

    The EtherType is a two-octet field in an Ethernet frame that defines the protocol encapsulated in the frame payload. If you do not configure this statement, all EtherTypes are reflected. Use an EtherType value that matches the EtherType value set on the customer premises equipment (CPE) to which your router connects. The EtherType value appears in the Ethernet type field of the packet. It specifies the protocol being transported in the Ethernet frame.

```
user@host# set reflect-etype ethertype-value
```

13. (Optional) Specify the reflection mode for the benchmarking test.

```
user@host# set reflect-mode (mac-swap | no-mac-swap)
```

Start and Stop the RFC 2544-Based Benchmarking Test

To start an RFC 2544-based benchmarking test:

- For Junos OS, issue the `test services rpm rfc2544-benchmarking test` *`test-name`* `start` CLI command.
- For Junos OS Evolved, issue the `test services monitoring rfc2544 test` *`test-name`* `start` CLI command.

To stop an RFC 2544-based benchmarking test:

- For Junos OS, issue the `test services rpm rfc2544-benchmarking test` *`test-name`* `stop` CLI command.
- For Junos OS Evolved, issue the `test services monitoring rfc2544 test` *`test-name`* `stop` CLI command.

To start an RFC 2544 benchmarking inet tests on Layer 3 VPN or virtual router:

- For Junos OS, issue the `test services rpm rfc2544-benchmarking test` *`test-name`* `routing-instance` *`routing-instance-name`* `start` CLI command.

- For Junos OS Evolved, issue the `test services monitoring rfc2544 test` *`test-name`* `routing-instance` *`routing-instance-name`* `start` CLI command.

To stop an RFC 2544 benchmarking inet tests on Layer 3 VPN or virtual router:

- For Junos OS, issue the `test services rpm rfc2544-benchmarking test` *`test-name`* `routing-instance` *`routing-instance-name`* `stop` CLI command.

- For Junos OS Evolved, issue the `test services monitoring rfc2544 test` *`test-name`* `routing-instance` *`routing-instance-name`* `stop` CLI command.

Copying an RFC 2544-Based Benchmarking Test Result

You can copy the RFC 2544-based benchmarking test results for a particular test ID to a local or a remote file.

- To copy test results to a local file:

    - For Junos OS, issue the `show services rpm rfc2544-benchmarking test-id` *`number`* `detail | save rfc-2544-test-result-session-id-`*`number`* CLI command.

    - For Junos OS Evolved, issue the `show services monitoring rfc2544 test-id` *`number`* `detail | save rfc-2544-test-result-session-id-`*`number`* CLI command.

- To copy test results to a remote file:

    - For Junos OS, issue the `show services rpm rfc2544-benchmarking test-id` *`number`* `detail | save ftp://username:password@sftpchannel.example.com/rfc-2544-test-result-session-id-`*`number`*.

    - For Junos OS Evolved, issue the `show services monitoring rfc2544 test-id` *`number`* `detail | save ftp://username:password@sftpchannel.example.com/rfc-2544-test-result-session-id-`*`number`*.

RELATED DOCUMENTATION

RFC 2544-Based Benchmarking Tests for ACX Routers Overview | 922

Layer 2 and Layer 3 RFC 2544-Based Benchmarking Test Overview | 928

show services rpm rfc2544-benchmarking

show services rpm rfc2544-benchmarking test-id

Configuring Ethernet Loopback for RFC 2544-Based Benchmarking Tests

Ethernet loopback is a feature that you can use for verifying the connectivity and identifying or isolating faults in a network.

On ACX Series routers, Ethernet loopback is supported on the egress user-to-network interfaces (UNIs) direction for a `bridge` family configuration. In ACX Series routers, Ethernet loopback is configured on the logical interfaces. The Ethernet loopback feature can be used in performance measurements where packets are looped back to the measuring device for testing various services.

Figure 74: Testing End-to-End Service in Ethernet Loopback Mode

Figure 74 on page 951 shows a scenario where UNI-B interface is configured in Ethernet loopback mode in the egress direction. The packets received on the network-to-network interface (NNI) of the ACX Series router are forwarded to the UNI-B interface and looped back at the UNI-B interface after the source and destination MAC addresses are swapped. This is a use case for testing an end-to-end service.

You can use the following optional parameters to identify an egress traffic flow for Ethernet loopback:

- Source MAC address
- Destination MAC address
- Source IPv4 address
- Destination IPv4 address
- VLAN
- VLAN .1p priority
- EtherType
- Test iterator duration

While performing RFC2544 benchmarking tests, configure Ethernet loopback as the test mode on a logical interface by including the `Ethernet-loopback` CLI statement at the `[edit services rpm rfc2544-benchmarking]` hierarchy level.

If you configure Ethernet loopback on logical interfaces without configuring any of the optional parameters, then any unknown unicast traffic in the same bridge domain also gets looped back and does not get forwarded to other logical interfaces while the test is being performed.

When an RFC2544 benchmarking test is being performed, if the **test-iterator-duration** parameter is not configured, then Ethernet loopback continues until the test is completed or terminated.

NOTE: When performing RFC2544 benchmarking tests, you can configure the test in initiator, reflector, or loopback mode. You cannot perform the RFC2544 benchmarking tests in a combination of these test modes.

The following is a sample Ethernet loopback configuration:

```
[edit services rpm rfc2544-benchmarking]
    tests {
        test-name test1{
            source-mac-address 00:bb:cc:dd:ee:ff;
            destination-mac-address 00:11:22:33:44:55;
            vlan-id 100;
            vlan-priority 2;
            vlan-cfi 1;
            ip-swap;
            udp-tcp-port-swap;
            forwarding-class network-control;
            packet-loss-priority medium-high;
            mode ethernet-loopback;
            family bridge;
            reflect-etype 2048;
            direction egress;
            source-udp-port 2020;
            destination-udp-port 3030;
            test-iterator-duration 50;
            test-interface ge-0/1/6.0;
        }
    }
[edit interfaces]
    ge-0/1/4 {
        flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
        unit 0 {
            encapsulation vlan-bridge;
            vlan-id 1000;
```

```
            family bridge {
                filter {
                    input ft1;
                }
            }
        }
    ge-0/1/6 {
        flexible-vlan-tagging;
        encapsulation flexible-ethernet-services;
        unit 0 {
            encapsulation vlan-bridge;
            vlan-id 100;
            input-vlan-map {
                push;
                vlan-id 1000;
            }
            output-vlan-map pop;
        }
    }
[edit routing-options]
    ppm {
        traceoptions {
            file ppmd size 100m;
            flag packet;
            flag event;
            flag distribute;
            flag pipe;
            flag all;
        }
    }
[edit firewall]
    family bridge {
        filter ft1 {
            term t1 {
                from {
                    user-vlan-id 100;
                }
                then count loopback;
            }
        }
    }
[edit bridge-domains]
    bd1 {
```

```
        interface ge-0/1/4.0;
        interface ge-0/1/6.0;
    }
```

RELATED DOCUMENTATION

RFC 2544-Based Benchmarking Test States

When you trigger an RFC 2544-based benchmarking test, it passes through a series of states. These states are displayed in the `Test state` field in the brief or detailed output of the `show services rpm rfc2544-benchmarking` command. The following are the names of the states through which the test progresses after it is initiated:

1. RFC2544_TEST_STATE_START_REQUEST—This is the first state that all the triggered tests enter. When a test enters this state, the state denotes that a request has been sent to a Packet Forwarding Engine to start the test.

2. RFC2544_TEST_STATE_START_FAILED—This state indicates that the test failed to start. This state occurs when the Packet Forwarding Engine responds to the **START_REQUEST** message. The Status field of the brief or detailed output of the `show` command displays a reason for the failure. When a test enters this state, it is categorized as an terminated test.

3. RFC2544_TEST_STATE_RUNNING—This state occurs if the Packet Forwarding Engine is able to successfully start the test. This state indicates that the test is in progress. You can use the output of the `show` command to learn additional information about the test progress.

4. RFC2544_TEST_STATE_STOP_REQUEST—A test enters this state when you use the `test services rpm rfc2544-benchmarking` *`test-id`* `stop` command. A request is sent to the Packet Forwarding Engine to stop the test.

5. RFC2544_TEST_STATE_STOP_FAILED—This state is entered when the Packet Forwarding Engine failed to stop a test after it received the **STOP_REQUEST** message. The Status field displays further information regarding the exact reason for failure.

6. RFC2544_TEST_STATE_STOPPED—This state is entered when the Packet Forwarding Engine successfully managed to stop a test when it received the **STOP_REQUEST** message.

7. RFC2544_TEST_STATE_COMPLETED—This state is entered when the test successfully completes all necessary test steps.

8. RFC2544_TEST_STATE_ABORTED_TIMEOUT—When a request is sent to the Packet Forwarding Engine for any test, a 10-second timer control is started. If a response is not received from the Packet Forwarding Engine and the timer elapses, the test is transitioned to the **ABORTED_TIMEOUT** state. This state is introduced to prevent a test from indefinitely waiting to receive a reply from the Packet Forwarding Engine.

9. RFC2544_TEST_STATE_RUNTIME_ERROR—This state is entered if the Packet Forwarding Engine encounters an error when the test is running. The Status field of the brief or detailed output specifies the reason for the failure. Tests that encounter the **RUNTIME_ERROR** state are added to the count of the terminated-tests category, which can be viewed from the output of the `show services rpm rfc2544-benchmarking` command.

RELATED DOCUMENTATION

RFC 2544-Based Benchmarking Tests for ACX Routers Overview | 922
Layer 2 and Layer 3 RFC 2544-Based Benchmarking Test Overview | 928
Configuring RFC 2544-Based Benchmarking Tests | 933
show services rpm rfc2544-benchmarking
show services rpm rfc2544-benchmarking test-id

Example: Configure an RFC 2544-Based Benchmarking Test for Layer 3 IPv4 Services

IN THIS SECTION

This example shows how to configure the benchmarking test for a Layer 3 IPv4 service.

> **NOTE**: This example is not applicable for ACX5448, ACX5048, and ACX5096 routers.

Requirements

This example uses the following hardware and software components:

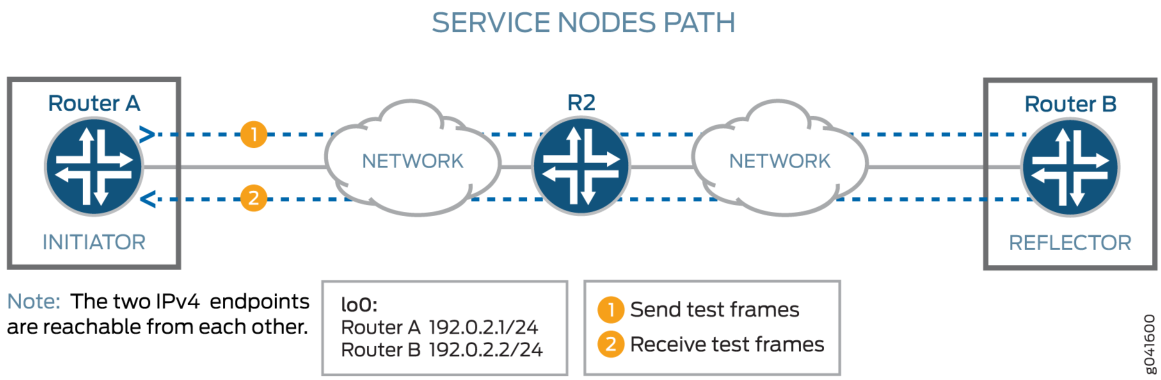
- An ACX Series router
- Junos OS Release 12.3X53 or later

Overview

Consider a sample topology in which a router, Router A, functions as an initiator and terminator of the test frames for an RFC 2544-based benchmarking test. Router A is connected over a Layer 3 network to another router, Router B, which functions as a reflector to reflect back the test frames it receives from Router A. IPv4 is used for transmission of test frames over the Layer 3 network. This benchmarking test is used to compute the IPv4 service parameters between Router A and Router B. Logical interfaces on both the routers are configured with IPv4 addresses to measure the performance attributes, such as throughput, latency, frame loss, and bursty frames, of network devices for the IPv4 service.

Figure 75 on page 957 shows the sample topology to perform an RFC 2544 test for a Layer 3 IPv4 service.

Figure 75: RFC 2544-Based Benchmarking Test for a Layer 3 IPv4 Service



Configuration

IN THIS SECTION

In this example, you configure the benchmarking test for a Layer 3 IPv4 service that is between interface ge-0/0/0 on Router A and interface ge-0/0/4 on Router B to detect and analyze the performance of the interconnecting routers. You do not configure a test profile on Router B, because it operates as a reflector. You must configure the reflector (Router B) before you configure the initiator (Router A), because the reflector needs to be already configured and the tests running before you start tests on the initiator. If you start the tests on the initiator first, then all the packets sent are lost until you start the tests on the reflector.

CLI Quick Configuration

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

Configure Benchmarking Test Parameters on Router B

```
set interfaces ge-0/0/4 unit 0 family inet address 192.0.2.2/24
set services rpm rfc2544-benchmarking tests test-name test1 test-interface ge-0/0/4.0
set services rpm rfc2544-benchmarking tests test-name test1 mode reflect
set services rpm rfc2544-benchmarking tests test-name test1 family inet
set services rpm rfc2544-benchmarking tests test-name test1 destination-ipv4-address 192.0.2.2
set services rpm rfc2544-benchmarking tests test-name test1 destination-udp-port 4001
set rfc2544-benchmarking tests test-name test1 source-ipv4-address 192.0.2.1
```

Configure Benchmarking Test Parameters on Router A

```
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.1/24
set rfc2544-benchmarking profiles test-profile throughput test-type throughput
set rfc2544-benchmarking profiles test-profile throughput packet-size 64
set rfc2544-benchmarking profiles test-profile throughput bandwidth-kbps 1000
set rfc2544-benchmarking tests test-name test1 test-profile throughput
set rfc2544-benchmarking tests test-name test1 test-interface ge-0/0/0.0
set rfc2544-benchmarking tests test-name test1 mode initiate-and-terminate
set rfc2544-benchmarking tests test-name test1 family inet
set rfc2544-benchmarking tests test-name test1 destination-ipv4-address 192.0.2.2
set rfc2544-benchmarking tests test-name test1 destination-udp-port 4001
set rfc2544-benchmarking tests test-name test1 source-ipv4-address 192.0.2.1
```

Configure Benchmarking Test Parameters on Router B

Step-by-Step Procedure

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

To configure the test parameters on Router B:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

   ```
   [edit]
   user@RouterB# edit interfaces
   ```

2. Configure the interface on which the test must be run.

```
[edit interfaces]
user@RouterB# edit ge-0/0/4
```

3. Configure a logical unit and specify the protocol family as `inet`.

```
[edit interfaces ge-0/0/4]
 user@RouterB# edit unit 0 family inet
```

4. Specify the address for the logical interface.

```
[edit interfaces ge-0/0/4 unit 0 family inet]
user@RouterB# set address 192.0.2.2/24
```

5. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit interfaces ge-0/0/4 unit 0 family inet]
user@RouterB# up
```

6. Go to the top level of the configuration mode.

```
[edit interfaces ge-0/0/4 unit 0]
user@RouterB# top
```

7. In configuration mode, go to the `[edit services]` hierarchy level.

```
[edit]
user@RouterB# edit services
```

8. Configure a real-time performance monitoring service (RPM) instance.

```
[edit services]
user@RouterB# edit rpm
```

9. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@RouterB# edit rfc2544-benchmarking
```

10. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@RouterB# edit tests test-name test1
```

11. Specify the logical interface, ge-0/0/4.0, on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# set test-interface ge-0/0/4.0
```

12. Specify `reflect` as the test mode for the packets that are sent during the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# set mode reflect
```

13. Configure the address type family, `inet`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# set family inet
```

14. Configure the destination IPv4 address for the test packets as 192.0.2.2. The destination IPv4 address configured on the reflector must match the destination IPv4 address configured on the initiator. If you configure 192.0.2.1 instead, you get this error message: `error: test test1 - Could not determine local interface for address 192.0.2.1`.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# set destination-ipv4-address 192.0.2.2
```

15. Specify the UDP port of the destination to be used in the UDP header for the generated frames as 4001.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# set destination-udp-port 4001
```

16. Configure the source IPv4 address for the test packets.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# set source-ipv4-address 192.0.2.1
```

17. Go to the top level of the configuration mode.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterB# top
```

18. Commit the configuration.

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

19. Confirm the configuration. If the output does not contain the configuration below, repeat the configuration instructions in this example to correct it.

```
[edit interfaces]
  ge-0/0/4 {
      unit 0 {
          family inet {
              address 192.0.2.2/24;
          }
       }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      # Note, When in reflector mode, test profile is not needed
      tests {
          test-name test1 {
```

```
                test-interface ge-0/0/4.0;
                mode reflect;
                family inet;
                destination-ipv4-address 192.0.2.2;
                destination-udp-port 4001;
                source-ipv4-address 192.0.2.1
            }
        }
    }
```

20. Exit to operational mode.

```
[edit]
user@RouterB# exit
user@RouterB>
```

21. Start the benchmarking test on the reflector.

```
user@RouterB> test services rpm rfc2544-benchmarking test test1 start
```

Once you configure the initiator (Router A), you can start the test on the initiator, and the initiator starts sending packets to the reflector. Once the test is successfully completed at the initiator, you can stop the test at the reflector by entering the `test services rpm rfc2544-benchmarking test test1 stop` command in operational mode.

Configure Benchmarking Test Parameters on Router A

Step-by-Step Procedure

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

To configure the test parameters on Router A:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

```
[edit]
user@RouterA# edit interfaces
```

2. Configure the interface on which the test must be run.

```
[edit interfaces]
user@RouterA# edit ge-0/0/0
```

3. Configure a logical unit and specify the protocol family.

```
[edit interfaces ge-0/0/0]
 user@RouterA# edit unit 0 family inet
```

4. Specify the address for the logical interface.

```
[edit interfaces ge-0/0/0 unit 0 family inet]
user@RouterA# set address 192.0.2.1/24
```

5. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit interfaces ge-0/0/0 unit 0 family inet]
user@RouterA# up
```

6. Go to the top level of the configuration command mode.

```
[edit interfaces ge-0/0/0 unit 0]
user@RouterA# top
```

7. In configuration mode, go to the `[edit services]` hierarchy level.

```
[edit]
user@RouterA# edit services
```

8. Configure a real-time performance monitoring service (RPM) instance.

```
[edit services]
user@RouterA# edit rpm
```

9. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@RouterA# edit rfc2544-benchmarking
```

10. Define a name for a test profile—for example, throughput.

```
[edit services rpm rfc2544-benchmarking]
user@RouterA# edit profiles test-profile throughput
```

11. Configure the type of test to be performed as throughput.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@RouterA# set test-type throughput
```

12. Specify the size of the test packet as 64 bytes.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@RouterA# set test-type packet-size 64
```

13. Define the theoretical maximum bandwidth for the test in kilobits per second, with a value from 1,000 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@RouterA# set test-type bandwidth-kbps 1000
```

14. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@RouterA# up
```

15. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles]
user@RouterA# up
```

16. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@RouterA# edit tests test-name test1
```

17. Specify the name of the test profile—for example, throughput—to be associated with a particular test name.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set test-profile throughput
```

18. Specify the logical interface, ge-0/0/0.0, on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set test-interface ge-0/0/0.0
```

19. Specify the test mode for the packets that are sent during the benchmarking test as initiate and terminate.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set mode initiate-and-terminate
```

20. Configure the address type family, `inet`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set family inet
```

21. Configure the destination IPv4 address for the test packets.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set destination-ipv4-address 192.0.2.2
```

22. Specify the UDP port of the destination to be used in the UDP header for the generated frames as 4001.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set destination-udp-port 4001
```

23. Configure the source IPv4 address for the test packets.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# set source-ipv4-address 192.0.2.1
```

24. Go to the top level of the configuration mode.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@RouterA# top
```

25. Commit the configuration.

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

26. Confirm the configuration. If the output does not contain the configuration below, repeat the configuration instructions in this example to correct it.

```
[edit]
user@RouterA# show
[edit interfaces]
  ge-0/0/0 {
      unit 0 {
          family inet {
              address 192.0.2.1/24;
          }
       }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      profiles {
```

```
        test-profile throughput {
            test-type throughput
            packet-size 64;
            bandwidth-kbps 1000;
        }
    }

    tests {
        test-name test1 {
            test-profile throughput;
            interface ge-0/0/0.0;
            mode initiate-and-terminate;
            family inet;
            destination-ipv4-address 192.0.2.2
            destination-udp-port 4001;
            source-ipv4-address 192.0.2.1
        }
    }
}
```

27. Exit to operational mode.

```
[edit]
user@RouterA# exit
user@RouterA>
```

28. Start the benchmarking test on the initiator.

```
user@RouterA> test services rpm rfc2544-benchmarking test test1 start
```

After the test successfully completes, it automatically stops at the initiator. Once the test is successfully completed at the initiator, you can stop the test at the reflector by entering the `test services rpm rfc2544-benchmarking test test1 stop` command on Router B in operational mode.

Results

If you have not done so already, confirm your configuration on Router A and Router B by entering the `show` command in configuration mode at the [edit interfaces] and [edit services rpm] hierarchy levels. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

Configuration for Benchmarking Test Parameters on Router A:

```
[edit interfaces]
  ge-0/0/0 {
      unit 0 {
          family inet {
              address 192.0.2.1/24;
          }
       }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      profiles {
          test-profile throughput {
              test-type throughput
              packet-size 64;
              bandwidth-kbps 1000;
          }
      }

      tests {
          test-name test1 {
              test-profile throughput;
              interface ge-0/0/0.0;
              mode initiate-and-terminate;
              family inet;
              destination-ipv4-address 192.0.2.2
              destination-udp-port 4001;
              source-ipv4-address 192.0.2.1
          }
      }
  }
```

Configuration for Benchmarking Test Parameters on Router B:

```
[edit interfaces]
  ge-0/0/4 {
      unit 0 {
          family inet {
```

```
            address 192.0.2.2/24;
        }
     }
}

[edit services rpm]
rfc2544-benchmarking {
    # Note, When in reflector mode, test profile is not needed
    tests {
        test-name test1 {
            test-interface ge-0/0/4.0;
            mode reflect;
            family inet;
            destination-ipv4-address 192.0.2.2;
            destination-udp-port 4001;
            source-ipv4-address 192.0.2.1
        }
    }
}
```

Verify the Results of the Benchmarking Test for Layer 3 IPv4 Services

IN THIS SECTION

Examine the results of the benchmarking test performed on the configured service between Router A and Router B.

Verify the Benchmarking Test Results

Purpose

Verify that the necessary and desired statistical values are displayed for the benchmarking test that is run on the configured service between Router A and Router B.

Action

In operational mode, enter the `show services rpm rfc2544-benchmarking (aborted-tests | active-tests | completed-tests | summary)` command, on either the initiator or the reflector, to display information about the results of each category or state of the RFC 2544-based benchmarking test, such as terminated tests, active tests, and completed tests, for each real-time performance monitoring (RPM) instance.

Meaning

The output displays the details of the benchmarking test that was performed.

RELATED DOCUMENTATION

Configuring RFC 2544-Based Benchmarking Tests | 933

rfc2544-benchmarking

profiles

tests

show services rpm rfc2544-benchmarking

show services rpm rfc2544-benchmarking test-id

Example: Configuring an RFC 2544-Based Benchmarking Test for NNI Direction of Ethernet Pseudowires

IN THIS SECTION

This example shows how to configure the benchmarking test for a network-to-network interface (NNI) direction of an Ethernet pseudowire service.

Requirements

This example uses the following hardware and software components:

- An ACX Series router
- Junos OS Release 12.3X52 or later

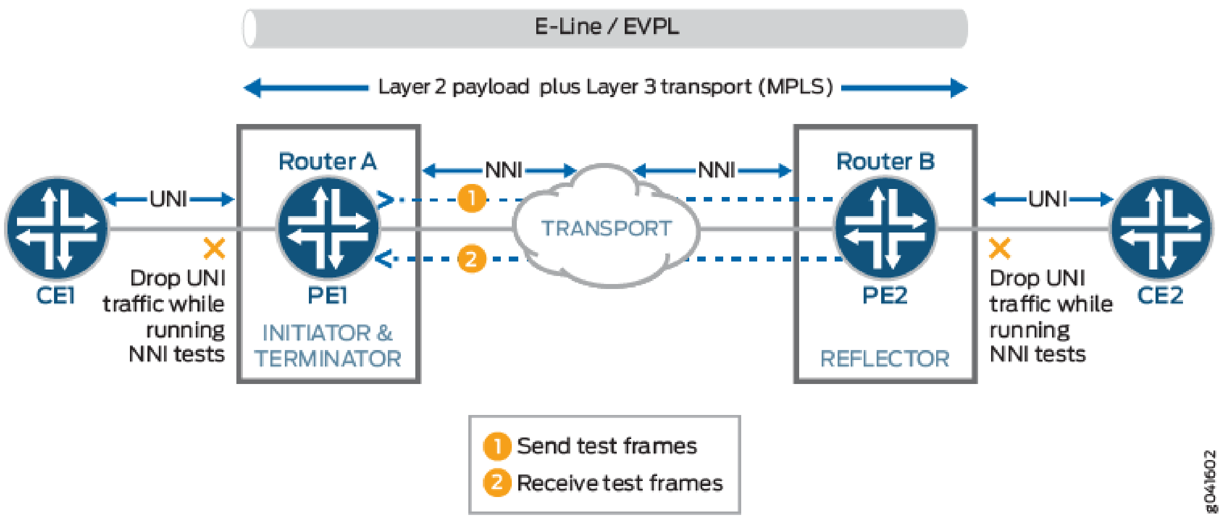
Overview

Consider a sample topology in which a router, Router A, functions as an initiator and terminator of the test frames for an RFC 2544-based benchmarking test. Router A operates as a provider edge device, PE1, which is connected to a customer edge device, CE1, on one side and over an Ethernet pseudowire to another router, Router B, which functions as a reflector to reflect back the test frames it receives from Router A. Router B operates as a provider edge device, PE2, which is the remote router located at the other side of the service provider core. The UNI direction of CE1 is connected to the NNI direction of PE1. An MPLS tunnel connects PE1 and PE2 over the Ethernet pseudowire or the Ethernet line (E-LINE).

This benchmarking test is used to compute the performance attributes in the network-to-network interface (NNI) direction of an Ethernet pseudowire service between Router A and Router B. The logical interface under test on Router A is the CE1 interface with UNI as the direction, and the logical interface under test on Router B is the CE2 interface with NNI as the direction. Data traffic arriving from UNI towards NNI is ignored while the test is in progress. Packets from NNI are not sent toward the customer edge because all packets are assumed to be test frames. The CCC family and NNI direction are configured on routers A and B.

Figure 76 on page 972 shows the sample topology to perform an RFC 2544 test for the NNI direction of an Ethernet pseudowire service.

Figure 76: RFC 2544-Based Benchmarking Test for NNI Direction of an Ethernet Pseudowire



Configuration

IN THIS SECTION

In this example, you configure the benchmarking test for the NNI direction of an Ethernet pseudowire service that is enabled between two routers to detect and analyze the performance of the interconnecting routers.

CLI Quick Configuration

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

Configuring Benchmarking Test Parameters on Router A

```
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 encapsulation vlan-ccc
set interfaces ge-0/0/0 unit 0 vlan-id 101
set services rpm rfc2544-benchmarking profiles test-profile throughput test-type throughput
set services rpm rfc2544-benchmarking profiles test-profile throughput packet-size 64
set services rpm rfc2544-benchmarking profiles test-profile throughput test-duration 20
set services rpm rfc2544-benchmarking profiles test-profile throughput bandwidth-kbps 500
set services rpm rfc2544-benchmarking tests test-name test1 interface ge-0/0/0.1
set services rpm rfc2544-benchmarking tests test-name test1 test-profile throughput
set services rpm rfc2544-benchmarking tests test-name test1 mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name test1 family ccc
set services rpm rfc2544-benchmarking tests test-name test1 direction nni
```

Configuring Benchmarking Test Parameters on Router B

```
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 0 encapsulation vlan-ccc
set interfaces ge-0/0/4 unit 0 vlan-id 101
set services rpm rfc2544-benchmarking tests test-name test1 interface ge-0/0/4.1
set services rpm rfc2544-benchmarking tests test-name test1 mode reflect
set services rpm rfc2544-benchmarking tests test-name test1 reflector-port 25
set services rpm rfc2544-benchmarking tests test-name test1 mode family ccc
set services rpm rfc2544-benchmarking tests test-name test1 direction uni
```

Configuring Benchmarking Test Parameters on Router B

Step-by-Step Procedure

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

To configure the test parameters on Router A:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

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

2. Configure the interface on which the test must be run.

```
[edit interfaces]
user@host# edit ge-0/0/0
```

3. Configure VLAN tagging for transmission and reception of 802.1Q VLAN-tagged frames.

```
[edit interfaces ge-0/0/0]
 user@host# set vlan-tagging
```

4. Configure a logical unit for the interface.

```
[edit interfaces ge-0/0/0]
 user@host# edit unit 0
```

5. Specify the encapsulation for Ethernet VLAN circuits.

```
[edit interfaces ge-0/0/0 unit 0]
 user@host# set encapsulation vlan-ccc
```

6. Configure the VLAN ID on the logical interface.

```
[edit interfaces ge-0/0/0 unit 0]
user@host# set vlan-id 101
```

7. Go to the top level of the configuration command mode.

```
[edit interfaces ge-0/0/0 unit 0]
user@host# top
```

8. In configuration mode, go to the `[edit services]` hierarchy level.

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

9. Configure a real-time performance monitoring service (RPM) instance.

```
[edit services]
user@host# edit rpm
```

10. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@host# edit rfc2544-benchmarking
```

11. Define a name for a test profile—for example, throughput.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit profiles test-profile throughput
```

12. Configure the type of test to be performed as throughput.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type throughput
```

13. Specify the size of the test packet as 64 bytes.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type packet-size 64
```

14. Specify the period—for example, 20 minutes—for which the test is to be performed in hours, minutes, or seconds by specifying a number followed by the letter h (for hours), m (for minutes), or s (for seconds).

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type test-duration 20m
```

15. Define the theoretical maximum bandwidth for the test in kilobits per second, with a value from 1 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type bandwidth-kbps 500
```

16. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# up
```

17. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles]
user@host# up
```

18. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

19. Specify the name of the test profile—for example, throughput—to be associated with a particular test name.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-profile throughput
```

20. Specify the logical interface, ge-0/0/0.1, on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-interface ge-0/0/0.1
```

21. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set mode initiate-and-terminate
```

22. Configure the address type family, `ccc`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set family ccc
```

23. Specify the direction of the interface on which the test must be run, which is NNI in this example.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set direction nni
```

Configuring Benchmarking Test Parameters on Router B

Step-by-Step Procedure

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

To configure the test parameters on Router B:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

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

2. Configure the interface on which the test must be run.

```
[edit interfaces]
user@host# edit ge-0/0/4
```

3. Configure VLAN tagging for transmission and reception of 802.1Q VLAN-tagged frames.

```
[edit interfaces ge-0/0/4]
 user@host# set vlan-tagging
```

4. Configure a logical unit for the interface.

```
[edit interfaces ge-0/0/4]
 user@host# edit unit 0
```

5. Specify the encapsulation for Ethernet VLAN circuits.

```
[edit interfaces ge-0/0/4 unit 0]
 user@host# set encapsulation vlan-ccc
```

6. Configure the VLAN ID on the logical interface.

```
[edit interfaces ge-0/0/4 unit 0]
user@host# set vlan-id 101
```

7. Go to the top level of the configuration command mode.

```
[edit interfaces ge-0/0/4 unit 0]
user@host# top
```

8. In configuration mode, go to the `[edit services]` hierarchy level.

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

9. Configure a real-time performance monitoring service (RPM) instance.

```
[edit services]
user@host# edit rpm
```

10. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@host# edit rfc2544-benchmarking
```

11. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

12. Specify the logical interface, ge-0/0/4.1, on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-interface ge-0/0/4.1
```

13. Specify `reflect` as the test mode for the packets that are sent during the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set mode reflect
```

14. Configure the address type family, `ccc`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set family ccc
```

15. Specify the direction of the interface on which the test must be run, which is NNI in this example.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set direction nni
```

Results

In configuration mode, confirm your configuration on Router A and Router B by entering the `show` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

Configuring Benchmarking Test Parameters on Router A:

```
[edit interfaces]
  ge-0/0/0 {
      vlan-tagging;
      unit 0 {
          encapsulation vlan-ccc;
          vlan-id 101;
      }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      profiles {
          test-profile throughput {
              test-type throughput
              packet-size 64;
              test-duration 20m;
              bandwidth-kbps 500;
          }
      }

      tests {
          test-name test1 {
              interface ge-0/0/0.1;
              test-profile throughput;
              mode initiate-and-terminate;
              family ccc;
              direction nni;
          }
      }
  }
```

Configuring Benchmarking Test Parameters on Router B:

```
[edit interfaces]
  ge-0/0/4 {
      vlan-tagging;
      unit 0 {
          encapsulation vlan-ccc;
          vlan-id 101;
      }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      # Note, When in reflector mode, test profile is not needed
      tests {
          test-name test1 {
              interface ge-0/0/4.1;
              mode reflect;
              family ccc;
              direction nni;
          }
      }
  }
```

After you have configured the device, enter the `commit` command in configuration mode.

Verifying the Results of the Benchmarking Test for NNI Direction of an Ethernet Pseudowire Service

IN THIS SECTION

Examine the results of the benchmarking test that is performed on the configured service between Router A and Router B.

Verifying the Benchmarking Test Results

Purpose

Verify that the necessary and desired statistical values are displayed for the benchmarking test that is run on the configured service between Router A and Router B.

Action

In operational mode, enter the `run show services rpm rfc2544-benchmarking (aborted-tests | active-tests | completed-tests | summary)` command to display information about the results of each category or state of the RFC 2544-based benchmarking test, such as terminated tests, active tests, and completed tests, for each real-time performance monitoring (RPM) instance.

Meaning

The output displays the details of the benchmarking test that was performed. For more information about the `run show services rpm rfc2544-benchmarking` operational command, see `show services rpm rfc2544-benchmarking` in the CLI Explorer.

RELATED DOCUMENTATION

rfc2544-benchmarking

profiles

tests

show services rpm rfc2544-benchmarking

show services rpm rfc2544-benchmarking test-id

Example: Configuring an RFC 2544-Based Benchmarking Test for UNI Direction of Ethernet Pseudowires

IN THIS SECTION

This example shows how to configure the benchmarking test for the user-to-network interface (UNI) direction of an Ethernet pseudowire service.

Requirements

This example uses the following hardware and software components:

- An ACX Series router
- Junos OS Release 12.3X53 or later

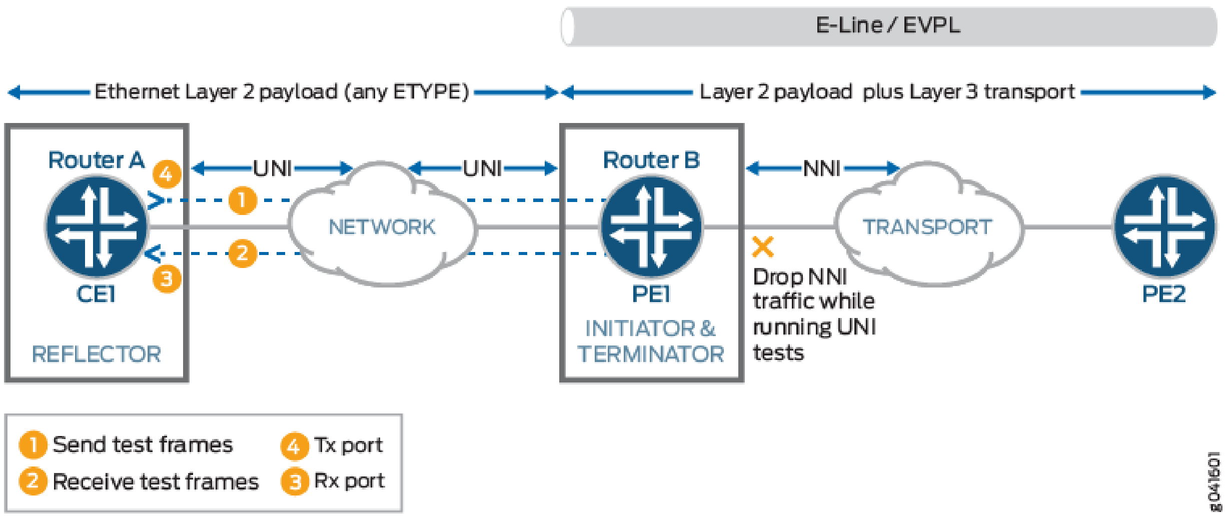
Overview

Consider a sample topology in which a router, Router A, functions as a reflector of the test frames for an RFC 2544-based benchmarking test. The logical customer edge (CE)-facing interface and `inet` family are configured on Router A. Router A is not part of a pseudowire and therefore, a Layer 3 family configuration is required on it. Router A, which is a customer edge device, CE1, is connected to Router B, which functions as a provider edge device, PE1, over an Ethernet pseudowire in the UNI direction with EtherType or Layer 2 Ethernet payload. The logical interface, CCC family, and UNI direction are configured on Router B. Router B or PE1 is connected over an Ethernet pseudowire in the NNI direction to a provider edge device at the remote site, PE2. The link between CE1 and PE1 is an Ethernet Layer 2 network and it can be configured with any EtherType value. The link between PE1 and PE2 is an Ethernet line (E-LINE) or an Ethernet Private Line (EPL) that has Layer 2 payload and Layer 3 transport sent over it. Router B or PE1 functions as an initiator and terminator of the test frames that are sent to Router A and reflected back from it.

This benchmarking test is used to compute the performance attributes in the user-to-network interface (UNI) direction of an Ethernet pseudowire service between Router A and Router B. Data traffic arriving from a network-to-network interface (NNI) toward the customer edge is ignored while the test is in progress. Packets from the CE are not sent toward the NNI because all packets are assumed to be test probes.

Figure 77 on page 984 shows the sample topology to perform an RFC 2544 test for the UNI direction of an Ethernet pseudowire service.

Figure 77: RFC 2544-Based Benchmarking Test for UNI Direction of an Ethernet Pseudowire



Configuration

IN THIS SECTION

In this example, you configure the benchmarking test for the UNI direction of an Ethernet pseudowire service that is enabled between two routers to detect and analyze the performance of the interconnecting routers.

CLI Quick Configuration

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

Configuring Benchmarking Test Parameters on Router A

```
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 0 vlan-id 101
set interfaces ge-0/0/0 unit 0 family inet address 200.0.0.1/24
set services rpm rfc2544-benchmarking profiles test-profile throughput test-type throughput
set services rpm rfc2544-benchmarking profiles test-profile throughput packet-size 64
set services rpm rfc2544-benchmarking profiles test-profile throughput test-duration 20m
set services rpm rfc2544-benchmarking profiles test-profile throughput bandwidth-kbps 500
set services rpm rfc2544-benchmarking tests test-name test1 interface ge-0/0/0.1
set services rpm rfc2544-benchmarking tests test-name test1 test-profile throughput
set services rpm rfc2544-benchmarking tests test-name test1 mode initiate-and-terminate
set services rpm rfc2544-benchmarking tests test-name test1 family inet
set services rpm rfc2544-benchmarking tests test-name test1 dest-address 10.200.0.2
set services rpm rfc2544-benchmarking tests test-name test1 udp-port 4001
```

Configuring Benchmarking Test Parameters on Router B

```
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 0 encapsulation vlan-ccc
set interfaces ge-0/0/4 unit 0 vlan-id 101
set services rpm rfc2544-benchmarking tests test-name test1 interface ge-0/0/4.1
set services rpm rfc2544-benchmarking tests test-name test1 mode reflect
set services rpm rfc2544-benchmarking tests test-name test1 mode family ccc
set services rpm rfc2544-benchmarking tests test-name test1 direction uni
```

Configuring Benchmarking Test Parameters on Router A

Step-by-Step Procedure

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

To configure the test parameters on Router A:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

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

2. Configure the interface on which the test must be run.

```
[edit interfaces]
user@host# edit ge-0/0/0
```

3. Configure VLAN tagging for transmission and reception of 802.1Q VLAN-tagged frames.

```
[edit interfaces ge-0/0/0]
 user@host# set vlan-tagging
```

4. Configure a logical unit and specify the protocol family as `inet`.

```
[edit interfaces ge-0/0/0]
 user@host# edit unit 0 family inet
```

5. Specify the address for the logical interface.

```
[edit interfaces ge-0/0/0 unit 0 family inet]
user@host# set address 10.200.0.1/24
```

6. Configure the VLAN ID on the logical interface as 101.

```
[edit interfaces ge-0/0/0 unit 0]
user@host# set vlan-id 101
```

7. Go to the top level of the configuration command mode.

```
[edit interfaces ge-0/0/0 unit 0]
user@host# top
```

8. In configuration mode, go to the `[edit services]` hierarchy level.

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

9. Configure a real-time performance monitoring service (RPM) instance.

```
[edit services]
user@host# edit rpm
```

10. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@host# edit rfc2544-benchmarking
```

11. Define a name for a test profile—for example, throughput.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit profiles test-profile throughput
```

12. Configure the type of test to be performed as throughput.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type throughput
```

13. Specify the size of the test packet as 64 bytes.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type packet-size 64
```

14. Specify the period for which the test is to be performed in hours, minutes, or seconds by specifying a number followed by the letter h (for hours), m (for minutes), or s (for seconds). In this example, you configure the period as 20 minutes.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type test-duration 20m
```

15. Define the theoretical maximum bandwidth for the test in kilobits per second, with a value from 1 Kbps through 1,000,000 Kbps.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# set test-type bandwidth-kbps 500
```

16. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles test-profile throughput]
user@host# up
```

17. Enter the `up` command to go the previous level in the configuration hierarchy.

```
[edit services rpm rfc2544-benchmarking profiles]
user@host# up
```

18. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

19. Specify the name of the test profile—for example, throughput—to be associated with a particular test name.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-profile throughput
```

20. Specify the logical interface, ge-0/0/0.1, on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-interface ge-0/0/0.1
```

21. Specify the test mode for the packets that are sent during the benchmarking test as initiation and termination.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set mode initiate-and-terminate
```

22. Configure the address type family, `inet`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set family inet
```

23. Configure the destination IPv4 address for the test packets as 200.0.0.2.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set dest-address 200.0.0.2
```

24. Specify the UDP port of the destination to be used in the UDP header for the generated frames as 4001.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set udp-port 4001
```

Configuring Benchmarking Test Parameters on Router B

Step-by-Step Procedure

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

To configure the test parameters on Router B:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level:

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

2. Configure the interface on which the test must be run.

```
[edit interfaces]
user@host# edit ge-0/0/4
```

3. Configure VLAN tagging for transmission and reception of 802.1Q VLAN-tagged frames.

```
[edit interfaces ge-0/0/4]
 user@host# set vlan-tagging
```

4. Configure a logical unit for the interface.

```
[edit interfaces ge-0/0/4]
 user@host# edit unit 0
```

5. Specify the encapsulation for Ethernet VLAN circuits.

```
[edit interfaces ge-0/0/4 unit 0]
 user@host# set encapsulation vlan-ccc
```

6. Configure the VLAN ID as 101 on the logical interface.

```
[edit interfaces ge-0/0/4 unit 0]
user@host# set vlan-id 101
```

7. Go to the top level of the configuration command mode.

```
[edit interfaces ge-0/0/4 unit 0]
user@host# top
```

8. In configuration mode, go to the `[edit services]` hierarchy level.

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

9. Configure a real-time performance monitoring service (RPM) instance.

```
[edit services]
user@host# edit rpm
```

10. Configure an RFC 2544-based benchmarking test for the RPM instance.

```
[edit services rpm]
user@host# edit rfc2544-benchmarking
```

11. Define a name for the test—for example, test1. The test name identifier can be up to 32 characters in length.

```
[edit services rpm rfc2544-benchmarking]
user@host# edit tests test-name test1
```

12. Specify the logical interface on which the RFC 2544-based benchmarking test is run.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set test-interface ge-0/0/4.1
```

13. Specify `reflect` as the test mode for the packets that are sent during the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set mode reflect
```

14. Configure the address type family, `ccc`, for the benchmarking test.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set family ccc
```

15. Specify the direction of the interface on which the test must be run, which is UNI in this example.

```
[edit services rpm rfc2544-benchmarking tests test-name test1]
user@host# set direction uni
```

Results

In configuration mode, confirm your configuration on Router A and Router B by entering the `show` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

Configuring Benchmarking Test Parameters on Router A:

```
[edit interfaces]
  ge-0/0/0 {
      vlan-tagging;
      unit 0 {
          vlan-id 101;
          family inet {
              address 10.200.0.1/24;
          }
      }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      profiles {
          test-profile throughput {
              test-type throughput
              packet-size 64;
              test-duration 20m;
              bandwidth-kbps 500;
          }
      }

      tests {
          test-name test1 {
              interface ge-0/0/0.1;
              test-profile throughput;
              mode initiate-and-terminate;
```

```
            family inet;
            dest-address 10.200.0.2
            udp-port 4001;
        }
    }
}
```

Configuring Benchmarking Test Parameters on Router B:

```
[edit interfaces]
  ge-0/0/4 {
      vlan-tagging;
      unit 0 {
          encapsulation vlan-ccc;
          vlan-id 101;
      }
  }

  [edit services rpm]
  rfc2544-benchmarking {
      # Note, When in reflector mode, test profile is not needed
      tests {
          test-name test1 {
              interface ge-0/0/4.1;
              mode reflect;
              family ccc;
              direction uni;
          }
      }
  }
```

After you have configured the device, enter the `commit` command in configuration mode.

Verifying the Results of the Benchmarking Test for UNI Direction of an Ethernet Pseudowire Service

IN THIS SECTION

Examine the results of the benchmarking test that is performed on the configured service between Router A and Router B.

Verifying the Benchmarking Test Results

Purpose

Verify that the necessary and desired statistical values are displayed for the benchmarking test that is run on the configured service between Router A and Router B.

Action

In operational mode, enter the `run show services rpm rfc2544-benchmarking (aborted-tests | active-tests | completed-tests | summary)` command to display information about the results of each category or state of the RFC 2544-based benchmarking test, such as terminated tests, active tests, and completed tests, for each real-time performance monitoring (RPM) instance.

Meaning

The output displays the details of the benchmarking test that was performed. For more information about the `run show services rpm rfc2544-benchmarking` operational command, see `show services rpm rfc2544-benchmarking` in the CLI Explorer.

RELATED DOCUMENTATION

Configuring RFC 2544-Based Benchmarking Tests | 933

rfc2544-benchmarking

profiles

tests

show services rpm rfc2544-benchmarking

show services rpm rfc2544-benchmarking test-id

Configuring a Service Package to be Used in Conjunction with PTP

On ACX1100 routers, you can configure a service package on the router for the RFC 2544-based benchmarking test, or for NAT and IPsec applications. When you configure the service package for the RFC 2544-based benchmarking test or for the NAT and IPsec applications, a reboot of the Forwarding Engine Board (FEB) occurs to apply the service package selection. By default, the service package for RFC 2544 benchmarking test is selected. The selection of a service package is needed on ACX1100 routers when you configure such routers for IEEE 1588v2 Precision Time Protocol (PTP) because both RFC 2544-based benchmarking tests and a combination of NAT and IPsec protocols are not supported simultaneously; you can configure only PTP and RFC 2544-based tests, or PTP and the combination of NAT and IPsec at a point in time.

You need to specify the service package to be RFC 2544-based or NAT and IPsec-based only for ACX1100-AC routers. The selection of a service package is not needed on ACX Series routers other than the ACX1100-AC and ACX500 routers because on such routers, only the RFC 2544-based benchmarking tests are supported; NAT and IPsec applications are not supported on those routers.

To configure the RFC 2544-based service package on a particular FPC, include the `service-package bundle-rfc2544` statement at the `[edit chassis fpc slot-number]` hierarchy level.

```
[edit chassis]
fpc slot-number {
    service-package bundle-rfc2544;
}
```

To configure the NAT and IPsec applications service package on a particular FPC, include the `service-package bundle-nat-ipsec` statement at the `[edit chassis fpc slot-number]` hierarchy level.

```
[edit chassis]
fpc slot-number {
    service-package bundle-nat-ipsec;
}
```

RELATED DOCUMENTATION

service-package

CHAPTER 19

Tracking Streaming Media Traffic Using Inline Video Monitoring

IN THIS CHAPTER

Understanding Inline Video Monitoring on MX Series Routers

SUMMARY

This topic gives an overview of inline video monitoring on Junos OS.

Junos OS supports inline video monitoring using media delivery index (MDI) metrics.

Use Feature Explorer: Inline Video Monitoring Services to confirm platform and release support for specific features. See *video-monitoring* for the configuration statement syntax.

Before you use the inline video monitoring feature, ensure that you understand the following terms:

- **Media delivery index**—MDI metrics facilitate identification of buffering needs for streaming media. Buffering must be adequate to compensate for packet jitter, measured by the MDI delay factor, and for quality problems indicated by lost packets, measured by the MDI media loss rate (MLR). By performing measurements under varying load conditions, you can identify sources of significant jitter or packet loss and take appropriate action.

- **Delay factor**—Delay factor is the maximum observed time difference between the arrival of media data and the drain of media data. The expected drain rate is the nominal, constant traffic rate for constant bit rate streams or the computed traffic rate of variable rate media stream packet data.

  For typical stream rates of 1 megabit per second and higher, an interval of one second provides an adequate sample time. The delay factor indicates how long a data stream must be buffered (delayed) at its nominal bit rate to prevent packet loss.

  The delay factor suggests the minimum size of the buffer required at the next downstream node. As a stream progresses, the variation of the delay factor indicates packet bunching or packet gaps (jitter). Greater delay factor values also indicate that more network latency is needed to deliver a stream due to the need to pre-fill a receive buffer before beginning the drain to guarantee no underflow.

  When the nominal drain bit rate at a receiving node is known, the delay factor's maximum indicates the size of buffer required to accommodate packet jitter.

- **Media rate variation (MRV)**—This value is the difference between the expected packet rate and actual packet rate, expressed as a percentage of the expected packet rate.

- **Media loss rate (MLR)**—This value is the number of media packets lost over a configurable time interval (*interval-duration*), where the flow packets carry streaming application information. A single IP packet can contain one or more streaming packets. For example, an IP packet typically contains seven 188-byte MPEG transport stream packets. In this case, a single IP packet loss results in seven lost packets counted (if those seven lost packets did not include null packets). Including out-of-order packets is important, because many consumer-type streaming devices do not attempt to reorder packets that are received out of order.

To configure the monitoring process, define criteria templates and apply them to the interfaces and flows you want to monitor. Monitoring templates include the following criteria:

- Duration of each measurement cycle

- Flow rate information used to establish expected flow rates

- Threshold levels for delay factor, media rate variation, and media loss rate that trigger desired system log alerts

For each interface you want to monitor, you can define one or more filters to select IPv4 flows for monitoring. Flows are designated as input or output flows. You can identify IPv4-over-MPLS flows, IPv6 flows, and IPv6-over MPLS flows. Starting in Junos OS Release 19.1R1, you can configure MX Series routers for inline video monitoring of uncompressed HD or 4K stream video (Payload Type 98 and 99). MDI functionality has been extended to video flows such as ST 2000-5 (RTP PT 98) and ST 2000-6 (RTP PT 99). These are non-MPEG video flows over IP/UDP/RTP and are constant bit-rate flows. The operator would specify proper IP addresses and UDP ports so that non-video flows over RTP will not go through MDI processing.

MPLS flows with more than three labels cannot be monitored.

IPv4 flows are uniquely identified by:

- Destination IP address
- Destination port
- Source IP address
- Source port
- Direction
- Interface index
- Media type (RTP or MPEG)

IPv4-over-MPLS flows are uniquely identified by:

- The top three MPLS labels
- Destination IP address
- Destination port
- Source IP address
- Source port
- Direction
- Interface index
- Media type (RTP or MPEG)

IPv6 flows are uniquely identified by:

- Destination IP address
- Destination port
- Source IP address
- Source port
- Direction
- Interface index
- Media type (RTP or MPEG)

IPv6-over-MPLS flows are uniquely identified by:

- The top three MPLS labels
- Destination IP address
- Destination port
- Source IP address
- Source port
- Direction
- Interface index
- Media type (RTP or MPEG)

Junos OS supports the definition of filters for up to 256 flows on an interface, which can consist of input flows, output flows, or a combination of input and output flows. These filters provide criteria for selecting flows for monitoring. If the selection criteria consist of lists of IP addresses or ports, you can exceed the maximum number of match conditions for flows. Video monitoring selects a widely variable number of flows based on flow filters.

The total number of destination IP addresses configured in a flow for an interface cannot exceed 32, and the total number of source IP addresses configured in a flow for an interface cannot exceed 32.

Inline video monitoring is not supported when you enable Next Gen Services on an MX Series router.

Inline video monitoring is available on MX Series 5G Universal Routing Platforms using only the following MPCs:

- MPC1
- MPC1E
- MPC2
- MPC2E
- MPC2E-NG
- MPC3E-NG
- MPC-16XGE
- MPC5E
- MPC6E

- MPC7E
- MPC8E
- MPC9E
- MPC10E
- MPC11E

The MX304 router and the MX Series routers that support the LC4800 and LC9600 line cards also support inline video monitoring.

> **NOTE**: Traffic throughput is reduced below the interface bandwidth when video monitoring is used with an MPC2E-NG or MPC3E-NG in the following scenario:
>
> - The input and output ports are on the same slot.
> - The input-flows is configured as inet and the output-flows is configured as mpls.
> - At least one flow has a traffic rate greater than 2 Gbps.
>
> To avoid this reduced throughput, use input and output ports on different slots.

You can configure the number of flows that can be measured per Packet Forwarding Engine at a time, up to a value of 8192. The maximum configured number of flows that can be measured for each MPC or line card model is shown in the second column of Table 141 on page 1000. The default number of flows that can be measured for each MPC or line card model is shown in the third column of Table 141 on page 1000.

When you do not define input or output flow filters for a monitored interface, all flows on the interface are subject to monitoring.

Table 141: MPC and Line Card Flow Monitoring Capacity by Model

Model	Maximum Configurable Number of Flows Monitored Simultaneously	Default Number of Flows Monitored Simultaneously
MPC1	8000	1000
MPC1E	8000	1000
MPC2	16,000	2000

Table 141: MPC and Line Card Flow Monitoring Capacity by Model *(Continued)*

Model	Maximum Configurable Number of Flows Monitored Simultaneously	Default Number of Flows Monitored Simultaneously
MPC2E	16,000	2000
MPC2E-NG	8000	1000
MPC3E-NG	8000	1000
MPC-16XGE	32,000	4000
MPC5E	40,000	5000
MPC6E	40,000	5000
MPC7E	40,000	5000
MPC8E	40,000	5000
MPC9E	40,000	5000
MPC10E	24,000 (starting in Junos OS Release 20.3R1)	3000
MPC11E	64,000 (starting in Junos OS Release 20.3R1)	8000
MX304 LMIC	24,000	3000
MX10K-LC4800	24,000	3000
MX10K-LC9600	48,000	6000

NOTE: Junos OS measures both UDP flows (the default) and RTP flows. Junos OS differentiates media traffic over UDP or RTP by inspecting the first byte in the UDP payload. If the first byte of the UDP payload is 0x47 (MPEG2-TS sync byte), the traffic is treated as media traffic over UDP. Traffic is treated as media traffic over RTP if the version field is 2 and the payload type is 33 in the RTP header. When neither of these criteria are met, the packet is not considered for video monitoring.

MX Series routers support the inline video monitoring to measure media delivery index (MDI) metrics that can be accessed using the SNMP GET operation. You use SNMP to monitor alarms raised by the inline video monitoring feature. The alarms are monitored in the network management systems either to troubleshoot the problem or to diagnose degradation in video quality.

You use the `video-monitoring` statement at the `[edit services]` hierarchy level to specify monitoring criteria for two key indicators of video traffic problems: delay factor and media loss rate (MLR), and to apply these metrics to flows on designated interfaces.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.3R2	Inline video monitoring is not supported when you enable Next Gen Services on an MX Series router.
19.1R1	You can configure MX Series routers for inline video monitoring of uncompressed HD or 4K stream video (Payload Type 98 and 99). MDI functionality has been extended to video flows such as ST 2000-5 (RTP PT 98) and ST 2000-6 (RTP PT 99). These are non-MPEG video flows over IP/UDP/RTP and are constant bit-rate flows. The operator would specify proper IP addresses and UDP ports so that non-video flows over RTP will not go through MDI processing.
17.4R1	You can identify IPv6 flows and IPv6-over MPLS flows.
17.2R1	You can identify IPv4-over-MPLS flows.
16.1R1	You can configure the number of flows that can be measured per Packet Forwarding Engine at a time, up to a value of 8192.
15.1R1	MX Series routers support the inline video monitoring to measure media delivery index (MDI) metrics that can be accessed using the SNMP GET operation. Currently, inline MDI can generate only a system log when the computed value is not within the configured range.

RELATED DOCUMENTATION

Configuring Inline Video Monitoring on MX Series Routers | 1003

show services video-monitoring mdi stats fpc-slot

show services video-monitoring mdi errors fpc-slot

show services video-monitoring mdi flows fpc-slot

alarms

Configuring Inline Video Monitoring on MX Series Routers

IN THIS SECTION

Configuring Media Delivery Indexing Criteria

To configure media delivery indexing criteria:

1. In edit mode, create a named template for video monitoring.

```
user@host# edit services video-monitoring templates template-name
```

   For example,

```
user@host# edit services video-monitoring templates t1
```

2. Set the duration for sampling in seconds. Flow media delivery indexing statistics are updated at the end of this interval.

```
[edit services video-monitoring templates template-name]
user@host# set interval-duration interval-duration
```

For example,

```
[edit services video-monitoring templates t1]
user@host# set interval-duration 1
```

BEST PRACTICE: If you change the interval duration when a template is being used, you cause a change in the calculated number of expected packets in an measurement interval for the template. We recommend that you do not change the interval duration for a template that is in use.

3. Set the inactivity timeout.

```
[edit services video-monitoring templates template-name]
user@host# set inactive-timeout inactive-timeout
```

For example,

```
[edit services video-monitoring templates t1]
user@host# set inactive-timeout 30
```

4. Configure either the media rate or layer 3 packet rate to establish expected flow rates used to compare to monitored flow rates.

NOTE: The media rate is the configured media bit rate for the stream. The media rate is used to establish *expected packets per second* (pps).

The Layer 3 packet rate in packets per second (pps) is used to establish *expected bits per second* (bps).

```
[edit services video-monitoring templates template-name]
user@host# set rate media media-bits-per-second
```

For example,

```
[edit services video-monitoring templates t1]
user@host# set rate media 2972400
```

5. Set delay factor thresholds for syslog message levels.

```
[edit services video-monitoring templates template-name]
user@host# set delay-factor threshold info delay-factor-threshold
user@host# set delay-factor threshold warning delay-factor-threshold
user@host# set delay-factor threshold critical delay-factor-threshold
```

For example,

```
[edit services video-monitoring templates t1]
user@host# set delay-factor threshold info 100
user@host# set delay-factor threshold warning 200
user@host# set delay-factor threshold critical 300
```

6. Set media loss rate thresholds for syslog message levels. You can set the threshold based on number of packets lost, or percentage of packets lost.

```
[edit services video-monitoring templates template-name]
user@host# set media-loss-rate threshold info percentage mlr-percentage
user@host# set media-loss-rate threshold warning percentage mlr-percentage
user@host# set media-loss-rate threshold critical percentage mlr-percentage
```

For example,

```
[edit services video-monitoring templates t1]
user@host# set media-loss-rate threshold info percentage 5
user@host# set media-loss-rate threshold warning percentage 10
user@host# set media-loss-rate threshold critical percentage 20
```

7. Set the media rate variation thresholds for syslog message levels. The threshold is based on the ratio of the *difference* between the configured media rate and the monitored media rate to the configured media rate, expressed as a percentage.

```
[edit services video-monitoring templates template-name]
user@host# set media-rate-variation threshold info mrv-variation
user@host# set media-rate-variation threshold warning mrv-variation
user@host# set media-rate-variation threshold critical mrv-variation
```

For example,

```
[edit services video-monitoring templates t1]
user@host# set media-rate-variation threshold info 10
user@host# set media-rate-variation threshold warning 15
user@host# set media-rate-variation threshold critical 20
```

Configuring Interface Flow Criteria

You can identify the input and output flows that you want to monitor. If you do not specify any identifiers, all flows on the interface are monitored. You can identify IPv4-over-MPLS flows. You can also identify IPv6 flows and IPv6-over MPLS flows. MPLS flows with more than three labels cannot be monitored.

NOTE: You can configure a maximum of 256 flow definitions for an interface. If your flow definitions contain lists of addresses and ports, you can exceed the number of match conditions. When you exceed the limits for flows or match conditions, you receive the following constraint message when you commit:

```
'interfaces xe-0/2/2.0'
    Number of flows or Number of match condition under flows exceeded limit
error: configuration check-out failed
```

To configure monitoring of flows for interfaces:

1. In edit mode, identify an interface for monitoring.

```
user@host# edit services video-monitoring interfaces interface-name
```

2. Identify IPv4 input flows for monitoring.

   a. Assign a name to the input flow.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set input-flows input-flow-name
```

b. Identify the source IP address or prefix value for the flow. You can use up to 32 addresses.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set input-flows input-flow-name source-address [ address ]
```

c. Identify the destination IP address or prefix value for the flow. You can use up to 32 addresses.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set input-flows input-flow-name destination-address [ address ]
```

d. Identify the source port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set input-flows input-flow-name source-port [ port ]
```

e. Identify the destination port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set input-flows input-flow-name destination-port [ port ]
```

f. Identify the template used to monitor the input flow on the interface.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set input-flows input-flow-name template template-name
```

3. Identify IPv4 output flows for monitoring.

a. Assign a name to the output flow.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set output-flows output-flow-name
```

b. Identify the source IP address or prefix value for the flow. You can use up to 32 addresses.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set output-flows output-flow-name source-address [ address ]
```

c. Identify the destination IP address or prefix value for the flow. You can use up to 32 addresses.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set output-flows output-flow-name destination-address [ address ]
```

d. Identify the source port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set output-flows output-flow-name source-port [ port ]
```

e. Identify the destination port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set output-flows output-flow-name destination-port [ port ]
```

f. Identify the template used to monitor the output flow on the interface.

```
[edit services video-monitoring interfaces interface-name family inet]
user@host# set output-flows output-flow-name template template-name
```

4. Identify IPv4-over-MPLS input flows for monitoring:

a. Assign a name to the input flow.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name
```

b. Identify the payload type as IPv4 over MPLS.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name payload-type ipv4
```

c. Identify the destination IP address or prefix value, the source IP address or prefix value, or both for the flow. You can use up to 32 destination addresses and up to 32 source addresses.

Identify the destination IP address or prefix value, the source IP address or prefix value, or both for the flow. You can use up to 32 destination addresses and up to 32 source addresses.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name destination-address [address]
user@host# set input-flows input-flow-name source-address [address]
```

If you configure multiple addresses for both the destination and source, then either all the destination or all the source values must have the same prefix length. For example, the following is allowed, because all the destination addresses have the same prefix length.

```
[edit services video-monitoring interfaces ge-0/2/2.0 family mpls]
user@host# set input-flows input-flow-name destination-address [203.0.13.0/24
198.51.100.0/24]
user@host# set input-flows input-flow-name source-address [172.16.0.0/12 192.0.2.11/32]
```

d. Identify the destination port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name destination-port [ port ]
```

e. Identify the source port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name source-port [ port ]
```

f. Identify the template used to monitor the input flow on the interface.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name template template-name
```

5. Identify IPv4-over-MPLS output flows for monitoring:

a. Assign a name to the output flow.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name
```

b. Identify the payload type as IPv4 over MPLS.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name payload-type ipv4
```

c. Identify the destination IP address or prefix value, the source IP address or prefix value, or both for the flow. You can use up to 32 destination addresses and up to 32 source addresses.

Identify the destination IP address or prefix value, the source IP address or prefix value, or both for the flow. You can use up to 32 destination addresses and up to 32 source addresses.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name destination-address [address]
user@host# set output-flows output-flow-name source-address [address]
```

If you configure multiple addresses for both the destination and source, then either all the destination or all the source values must have the same prefix length. For example, the following is allowed, because all the destination addresses have the same prefix length.

```
[edit services video-monitoring interfaces ge-0/2/2.0 family mpls]
user@host# set output-flows output-flow-name destination-address [203.0.13.0/24
198.51.100.0/24]
user@host# set output-flows output-flow-name source-address [172.16.0.0/12 192.0.2.11/32]
```

d. Identify the source port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name source-port [port]
```

e. Identify the destination port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name destination-port [ port ]
```

f. Identify the template used to monitor the output flow on the interface.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name template template-name
```

6. Identify IPv6 input flows for monitoring.

a. Assign a name to the input flow.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set input-flows input-flow-name
```

b. Identify the source IP address or prefix value for the flow. You can use up to 32 addresses.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set input-flows input-flow-name source-address [ address ]
```

c. Identify the destination IP address or prefix value for the flow. You can use up to 32 addresses.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set input-flows input-flow-name destination-address [ address ]
```

d. Identify the source port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set input-flows input-flow-name source-port [ port ]
```

e. Identify the destination port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set input-flows input-flow-name destination-port [ port ]
```

f. Identify the template used to monitor the input flow on the interface.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set input-flows input-flow-name template template-name
```

7. Identify IPv6 output flows for monitoring.

   a. Assign a name to the output flow.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set output-flows output-flow-name
```

   b. Identify the source IP address or prefix value for the flow. You can use up to 32 addresses.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set output-flows output-flow-name source-address [ address ]
```

   c. Identify the destination IP address or prefix value for the flow. You can use up to 32 addresses.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set output-flows output-flow-name destination-address [ address ]
```

   d. Identify the source port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set output-flows output-flow-name source-port [ port ]
```

   e. Identify the destination port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set output-flows output-flow-name destination-port [ port ]
```

   f. Identify the template used to monitor the output flow on the interface.

```
[edit services video-monitoring interfaces interface-name family inet6]
user@host# set output-flows output-flow-name template template-name
```

8. Identify IPv6-over-MPLS input flows for monitoring:

a. Assign a name to the input flow.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name
```

b. Identify the payload type as IPv6 over MPLS.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name payload-type ipv6
```

c. Identify the destination IP address or prefix value, the source IP address or prefix value, or both for the flow. You can use multiple addresses (up to 32) for either the destination or the source IP address, but not for both.

Identify the destination IP address or prefix value, the source IP address or prefix value, or both for the flow. You can use multiple addresses (up to 32) for either the destination or the source IP address, but not for both.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name destination-address [address]
user@host# set input-flows input-flow-name source-address [address]
```

d. Identify the destination port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name destination-port [ port ]
```

e. Identify the source port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name source-port [ port ]
```

f. Identify the template used to monitor the input flow on the interface.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set input-flows input-flow-name template template-name
```

9. Identify IPv6-over-MPLS output flows for monitoring:

a. Assign a name to the output flow.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name
```

b. Identify the payload type as IPv6 over MPLS.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name payload-type ipv6
```

c. Identify the destination IP address or prefix value, the source IP address or prefix value, or both for the flow. You can use multiple addresses (up to 32) for either the destination or the source IP address, but not for both.

Identify the destination IP address or prefix value, the source IP address or prefix value, or both for the flow. You can use multiple addresses (up to 32) for either the destination or the source IP address, but not for both.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name destination-address [address]
user@host# set output-flows output-flow-name source-address [address]
```

d. Identify the source port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name source-port [port]
```

e. Identify the destination port for the flow. You can use multiple port numbers and port ranges.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name destination-port [ port ]
```

f. Identify the template used to monitor the output flow on the interface.

```
[edit services video-monitoring interfaces interface-name family mpls]
user@host# set output-flows output-flow-name template template-name
```

Configuring the Number of Flows That Can Be Measured

You can configure the number of flows that can be measured per Packet Forwarding Engine at a given time by an MPC. This value takes effect the next time the MPC is rebooted. If you do not configure this value, the default maximum value for an MPC is given in "Understanding Inline Video Monitoring on MX Series Routers" on page 996.

To configure the number of flows that can be measured per Packet Forwarding Engine by an MPC at a given time:

- Configure the flow table size. The range is 16 through 8192.

```
[edit chassis fpc slot inline-video-monitoring]
user@host# set flow-table-size size
```

RELATED DOCUMENTATION

Understanding Inline Video Monitoring on MX Series Routers | 996

templates

interfaces

Inline Video Monitoring Syslog Messages on MX Series Routers

The following examples show the syslog messages produced when configured video monitoring thresholds are exceeded.

/var/log/messages

```
Mar 11 18:36:25  tstrtr01 fpc2 [MDI] DF: 56.71 ms, exceeded threshold for flow(src:192.0.2.2
dst:198.51.100.2 sport:1024 dport:2048) ingressing at interface xe-2/2/1.0 with template t1.
Mar 11 18:36:25  tstrtr01 fpc2 [MDI] MLR : 112, exceeded threshold for flow (src:192.0.2.2
dst:198.51.100.2 sport:1024 dport:2048) ingressing at interface xe-2/2/1.0 with template t1.
Mar 11 18:36:25  tstrtr01 fpc2 [MDI] MRV : -5.67, exceeded threshold for flow (src:192.0.2.2
dst:198.51.100.2 sport:1024 dport:2048) ingressing at interface xe-2/2/1.0 with template t1.
```

Console Messages

```
NPC2(tstrtr01 vty)# [Mar 12 01:40:58.411 LOG: Critical] [MDI] MLR : 420, exceeded threshold for
flow (src:192.0.2.2 dst:198.51.100.2 sport:1024 dport:2048) ingressing at interface xe-2/2/1.0
with template t1.
[Mar 12 01:40:58.411 LOG: Critical] [MDI] MRV : -14.89, exceeded threshold for flow
(src:192.0.2.2 dst:198.51.100.2 sport:1024 dport:2048) ingressing at interface xe-2/2/1.0 with
template t1.
[Mar 12 01:40:59.412 LOG: Critical] [MDI] DF: 141.74 ms, exceeded threshold for
flow(src:192.0.2.2 dst:198.51.100.2 sport:1024 dport:2048) ingressing at interface xe-2/2/1.0
with template t1.
```

RELATED DOCUMENTATION

Configuring Inline Video Monitoring on MX Series Routers | 1003

Generation of SNMP Traps and Alarms for Inline Video Monitoring on MX Series Routers

IN THIS SECTION

SNMP support is introduced for the Media Delivery Index (MDI) metrics of inline video monitoring. Inline video monitoring is available on MX Series routers using only MPCE1, MPC2, MPC2E, MPC2E-NG, MPC5E, MPC6E, MPC7E, MPC8E, and MPC- 16XGE. Starting in Junos OS Release 20.3R1, inline video monitoring is available on MX Series routers using MPC10E and MPC11E.

SNMP traps are triggered when the computed delay factor (DF), media rate variation (MRV), and media loss rate (MLR) value are not within the configured range. You can retrieve the MDI statistics, flow levels, error details, and MDI record-level information using SNMP Get and Get Next requests. The

SNMP traps and alarms that are generated when the MDI metrics exceed the configured ranges can be cleared as necessary. Also, you can control the flooding of SNMP traps on the system.

The following sections describe the statistical counters and parameters that are collected for MDI records and for generation of SNMP traps and alarms when the DF, MRV, and MLR values are not within the specified ranges.

Collection of MDI Statistics Associated with an FPC Slot

The FPC-level statistics include the following parameters that are displayed in the output of the `show services video-monitoring mdi stats fpc-slot` *`fpc-slot`* command. All of these attributes can be obtained using the SNMP Get request.

Table 142: show services video-monitoring mdi stats fpc-slot Output Fields

Field Name	Field Description
`FPC Slot`	Slot number of the monitored FPC
`Active Flows`	Number of active flows currently monitored. active flows = inserted flows - deleted flows.
`Total Inserted Flows`	Number of flows initiated under video monitoring.
`Total Deleted Flows`	Number of flows deleted due to inactivity timeout.
`Total Packets Count`	Number of total packets monitored.
`Total Bytes Count`	Number of total bytes monitored.
`DF Alarm Count`	Number of delay factor alarms at each of the following levels: • `Info level` • `Warning level` • `Critical level`

Table 142: show services video-monitoring mdi stats fpc-slot Output Fields *(Continued)*

Field Name	Field Description
`MLR Alarm Count`	Number of media loss rate (MLR) alarms at each of the following levels: • `Info level` • `Warning level` • `Critical level`
`MRV alarm count`	Number of media rate variation (MRV) alarms at each of the following levels: • `Info level` • `Warning level` • `Critical level`

Collection of MDI Errors Associated with an FPC Slot

The FPC-level statistics include the following parameters that are displayed in the output of the `show services video-monitoring mdi errors fpc-slot` *`fpc-slot`* command. All of these attributes can be obtained using the SNMP Get request.

Table 143: show services video-monitoring mdi errors fpc-slot Output Fields

Field Name	Field Description
`FPC slot`	Slot number of the monitored FPC.
`Flow Insert Error`	Number of errors during new flow insert operations.
`Flow Policer Drops`	Number of packets dropped by flow policer process. **NOTE**: New flows usually arrive within a very short time interval (1.5 microseconds). These errors do not represent the loss of entire flows, because subsequent packets in the flow can establish the flow. All packets are monitored after a flow has been established. Packet forwarding occurs independently of the video monitoring, and packets are not dropped due to video monitoring errors.

Table 143: show services video-monitoring mdi errors fpc-slot Output Fields *(Continued)*

Field Name	Field Description
`Unsupported Media Packets Count`	Number of packets dropped because they are not media packets or they are unsupported media packets.
`PID Limit Exceeded`	Number of packets unmonitored because the process identifier (PID) limit exceeded has been exceeded. **NOTE**: The current PID limit is 6.

Collection of MDI Flows Associated with an FPC Slot

The FPC-level statistics include the following parameters that are displayed in the output of the `show services video-monitoring mdi flows fpc-slot` *`fpc-slot`* command. All of these attributes can be obtained using the SNMP Get request.

Table 144: show services mdi flows Output Fields

Field Name	Field Description
`SIP`	Source IP address
`DIP`	Destination IP address
`SP`	Source port
`DP`	Destination port
`Di`	Direction (I=Input, O=Output)
`Ty`	Type of flow
`Last DF:MLR`	Delay factor and media loss rate value of last media delivery index record

Table 144: show services mdi flows Output Fields *(Continued)*

Field Name	Field Description
`Avg DF:MLR`	Average value of delay factor and media loss rate
`Last MRV`	Media rate variation value of last media delivery index record
`Avg MRV`	Average value of media rate variation
`IFL`	Interface name on which flow is receiving
`Template Name`	Name of template associated with flow

Collection of MDI Record-Level Metrics

The computed DF, MLR, and MRV counters of all valid MDI records of a flow that you can view by using the output of the `show services video-monitoring mdi flows fpc-slot` *`fpc-slot`* `detail` command can be obtained by using the SNMP Get request.

RELATED DOCUMENTATION

Understanding Inline Video Monitoring on MX Series Routers | 996

SNMP Traps for Inline Video Monitoring Statistics on MX Series Routers

SNMP is primarily used to monitor alarms raised by the inline video monitoring feature. The alarms sent to a network management system (NMS) either to troubleshoot the problem quickly or to proactively diagnose degradation in video quality. The following SNMP traps or alarms are implemented with the Cleared, Info, Warning, and Critical severity levels. The Cleared severity level is used to indicate a normal condition and to clear a particular alarm. Whenever a change in the alarm level occurs, the corresponding alarm is generated.

All the alarms include the following information pertaining to the MDI flows:

- Source IP address

- Destination IP address
- Source Port Destination
- Port Traffic type (UDP or RTP)
- Computed DF, MLR, and MRV values

The following traps are generated for MDI metrics:

- mdiMLRAlarm—This trap is generated when the computed MLR value is not within the configured range.
- mdiDFAlarm—This trap is generated when the computed DF value is not within the configured range.
- mdiMRVAlarm—This trap is generated when the computed MRV value is not within the configured range.

To enable the generation of SNMP traps or alarms for inline video monitoring or MDI metrics, include the `alarms` statement and its substatements at the `[edit services video-monitoring]` hierarchy level.

Processing SNMP GET Requests for MDI Metrics on MX Series Routers

A query on-demand mechanism without caching facility is used to process the SNMP Get requests. The Routing Engine queries the Packet Forwarding Engine to obtain the computed metrics on every Get request. The Routing Engine does not maintain computed metrics locally. No additional memory is required to cache queried metrics. The network management system (NMS) server can receive latest information on every Get request, especially regarding the MDI records because MDI records are updated very frequently. However, querying the Packet Forwarding Engine PFE on each GET request is resource-consuming if the volume of metrics is large. The response to a Get request might be relatively delayed as the Routing Engine has to poll the Packet Forwarding Engine to obtain the metrics.

Inline MDI metrics are real-time data where cached information might not be valid. Reporting cached or invalid metrics is not beneficial because it a real-time monitoring feature. An increase in the number of flows and number of MDI records per flow causes a proportional increase in the volume of memory required in the Routing Engine to store flows and MDI records for all flows. Because asynchronous traps are generated for threshold with enough contents, frequent Get request from NMS are not highly expected, reducing the periodicity of polling to the Packet Forwarding Engine. SNMP traps are triggered with the severity level of Info, Warning, Critical, or Cleared. A trap with the cleared severity level is used to clear an alarm.

Whenever a change in the alarm level occurs, the designated trap is triggered. For example, if the delay factor (DF) alarm changes from informational level to warning level, or from warning to critical, the mdiDFAlarm trap is triggered. Alarm can be immediate or average. If the immediate alarm is configured, an immediate trap is raised at the end of interval duration if the metric value exceeds the configured range. If the average alarm is configured, a trap is generated, based on the average value for specified number of interval duration.

Storm control is applied for SNMP traps at the flow level and not at the FPC level. The NMS system can obtain SNMP trap from all the flows even if multiple flows are generating traps at approximately the same time. If multiple flows are generating traps at nearly the same time, NMS is flooded by many traps at the same time. For example, no traffic received on a logical interface owing to any reason can trigger all alarms and cause an avalance of alarms on the NMS server.

RELATED DOCUMENTATION

Understanding Inline Video Monitoring on MX Series Routers | **996**

6

PART

Configuration Statements and Operational Commands

Junos CLI Reference Overview

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

- Junos CLI Reference

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

- Configuration Statements
- Operational Commands