JUNIPER NETWORKS

Junos® OS

Flow Monitoring Feature Guide

Modified: 2017-02-06

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

Junos® OS Flow Monitoring Feature 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

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

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

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

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

Supported Platforms

For the features described in this document, the following platforms are supported:

- MX Series
- T Series
- M Series
- PTX Series

Using the Examples in This Manual

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

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

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

Merging a Full Example

To merge a full example, follow these steps:

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

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

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

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

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

Merging a Snippet

To merge a snippet, follow these steps:

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

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

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


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

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

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

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

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

Documentation Conventions

Table 1 on page xiii defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xiv defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		static {
		route default {
		nexthop *address*;
		retain;
		}
		}
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

GUI Conventions

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

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

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

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

PART 1

Overview

CHAPTER 1

Understanding Flow Monitoring

Flow Monitoring Overview

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

Using a Juniper Networks router, a selection of PICs for M Series and T Series routers—including the Monitoring Services PIC, Monitoring Services II PIC, Adaptive Services PIC, and MultiServices PICs—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 traffic flows between source and destination routers in your network.
- Sample all incoming traffic on the monitoring interface and present the data in record format.
- Encrypt or tunnel outgoing records, intercepted traffic, or both.
- Direct filtered traffic to different packet analyzers and present the data in its original format.
- Intercept unwanted traffic, discard it, and perform accounting on the discarded packets.

There are two main types of flow monitoring:

- Active Flow Monitoring
- Passive Flow Monitoring

Related Documentation

- Active Flow Monitoring Overview on page 91
- Passive Flow Monitoring Overview on page 29

- *Active Flow Monitoring Overview*
- *Passive Flow Monitoring Overview*

Terms and Acronyms

A

active flow monitoring
Technique to lawfully intercept and observe specified data network traffic on an active router participating in the network.

Adaptive Services PIC
Advanced PIC that handles active flow monitoring, Network Address Translation (NAT), stateful firewall, and intrusion detection functions. For more information on the Adaptive Services PIC, see the *Junos Services Interfaces Configuration Guide*.

C

cflowd
Version 5 and version 8 flow monitoring process that captures flow information from network traffic and exports this data into summary tables. Once captured, flow data can be analyzed as needed. For more information about cflowd, see **http://www.caida.org**.

content destination
A recipient of monitored packets sent by a DTCP or dynamic flow capture-enabled monitoring station.

control source
A dynamic flow capture client that wants to monitor electronic data or voice transfer over the network. The control source sends filter requests to the dynamic flow capture-enabled monitoring station by using DTCP.

D

DTCP (Dynamic Tasking Control Protocol)
Protocol used to specify filtering criteria in a dynamic flow capture environment.

dynamic flow capture
Technique that allows 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.

E

ES PIC
PIC that handles encryption and security services (such as IP Security [IPSec]).

F

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

M

Monitoring Services II PIC
Advanced PIC that handles passive flow monitoring functions.

Monitoring Services III PIC Advanced PIC that handles dynamic flow capture functions.

Monitoring Services PIC Original PIC that handles passive and active flow monitoring functions.

MultiServices 100 PIC Also referred to as MultiServices PIC Type 1. Advanced PIC that handles active flow capture functions.

MultiServices 400 PIC Also referred to as MultiServices PIC Type 2. Advanced PIC that handles active flow capture functions.

MultiServices 500 PIC Also referred to as MultiServices PIC Type 3. Advanced PIC that handles active flow capture functions.

P

passive flow monitoring Technique to lawfully intercept and observe specified data network traffic on a passive flow monitoring station not participating in the network.

More Information About Flow Monitoring

To learn more about passive flow monitoring, active flow monitoring, cflowd versions 5 and 8, and flow monitoring version 9 see the following:

- Version 9: RFC 3954 at http://www.faqs.org/rfcs/rfc3954.html
- Versions 5 and 8: Cooperative Association for Internet Data Analysis (CAIDA) website at http://www.caida.org
- *Junos Services Interfaces Configuration Guide*
- *Junos Policy Framework Configuration Guide*
- Internet draft draft-cavuto-dtcp-01.txt, *DTCP: Dynamic Tasking Control Protocol* (expires March 2007)

For more information on IPSec and the ES PIC, see the *Junos System Basics Configuration Guide*.

CHAPTER 2

Understanding Flow Monitoring Output Formats

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:

- Version 5 Formats and Fields on page 8

- Version 8 Formats and Fields on page 11
- Version 9 Formats and Fields on page 18

Version 5 Formats and Fields

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

- Figure 1 on page 8
- Table 3 on page 8
- Figure 2 on page 9
- Table 4 on page 9

Figure 1: Version 5 Packet Header Format

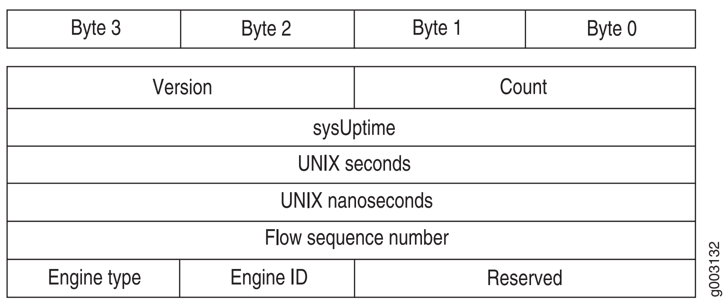


Table 3: Export Version 5 Packet Header Fields

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
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

Table 3: Export Version 5 Packet Header Fields *(continued)*

Field	Description	Comments
Engine ID	User-configured 8-bit value	–

Figure 2: Version 5 Flow-Export Flow Header Format

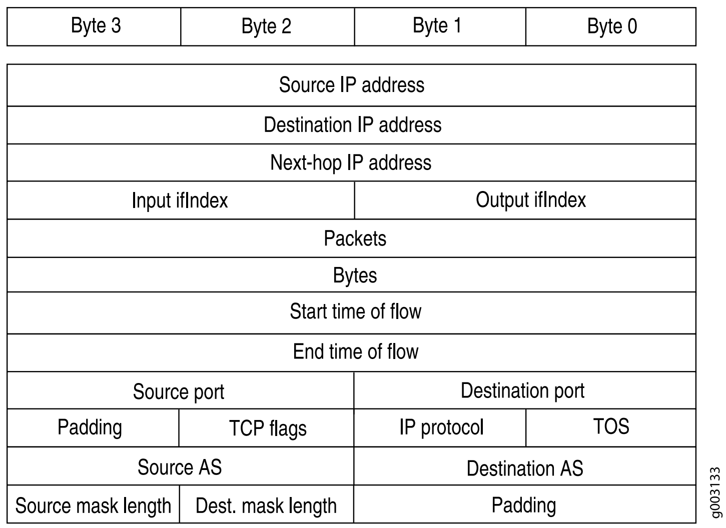


Table 4: Export Version 5 Flow-Export Flow Header Fields

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	Junos OS Release 5.7 and later—Dynamically inserted, but overridden by manual configuration Junos OS Release 5.5—Manually set Junos OS Release 5.4—Set to zero
Output ifIndex	SNMP index value for the output interface where the router forwards flows	Junos OS Release 5.7 and later—Dynamically inserted, but overridden by manual configuration Junos OS Release 5.5—Manually set Junos OS Release 5.4—Set to zero

Table 4: Export Version 5 Flow-Export Flow Header Fields *(continued)*

Field	Description	Comments
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
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	Junos OS Release 5.7 and later—Dynamically inserted if AS information is available
Destination AS	AS number of the destination address	Junos OS Release 5.7 and later—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).

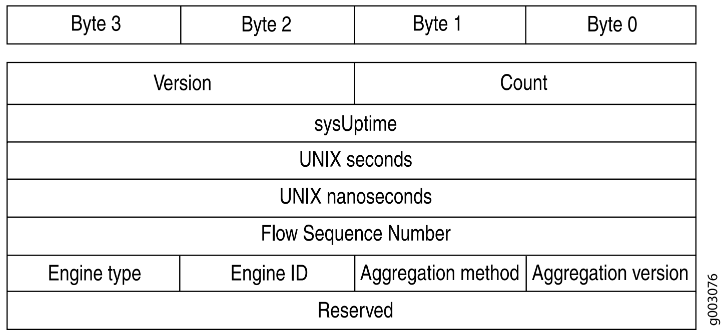
For more information on ICMP type and type code, see RFC 792 at **http://www.ietf.org**.

Version 8 Formats and Fields

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

- Figure 3 on page 12
- Table 5 on page 12
- Figure 4 on page 13
- Table 6 on page 13
- Figure 5 on page 13
- Table 7 on page 14
- Figure 6 on page 15
- Table 8 on page 15
- Figure 7 on page 16
- Table 9 on page 16
- Figure 8 on page 17
- Table 10 on page 17

Figure 3: Version 8 Template Flow Format



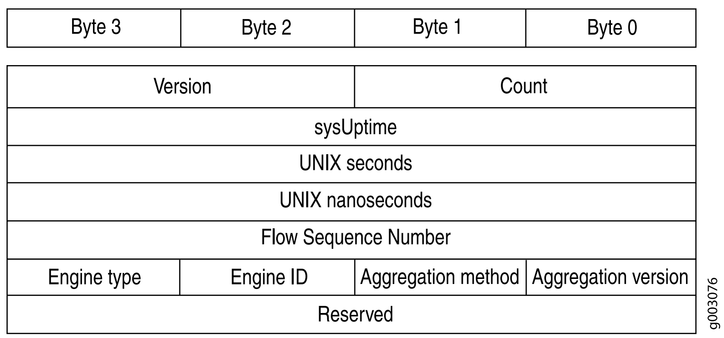


Table 5: Version 8 Flow Template Fields

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
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

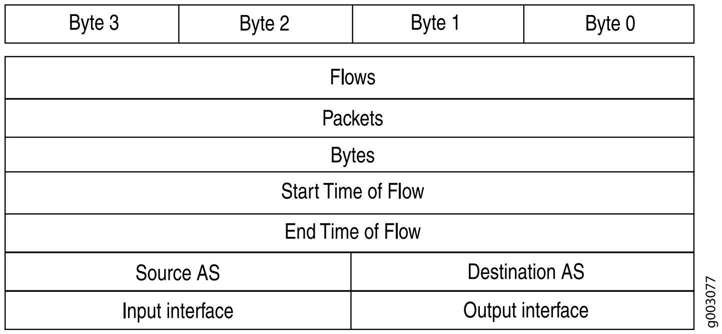


Table 6: Version 8 AS Aggregation Flow Entry Fields

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

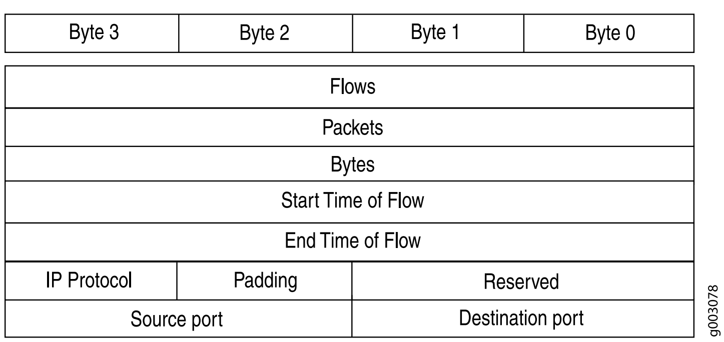


Table 7: Version 8 Protocol/Port Aggregation Flow Entry Fields

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
Destination port	Destination application port

Figure 6: Version 8 Prefix Aggregation Flow Entry Format

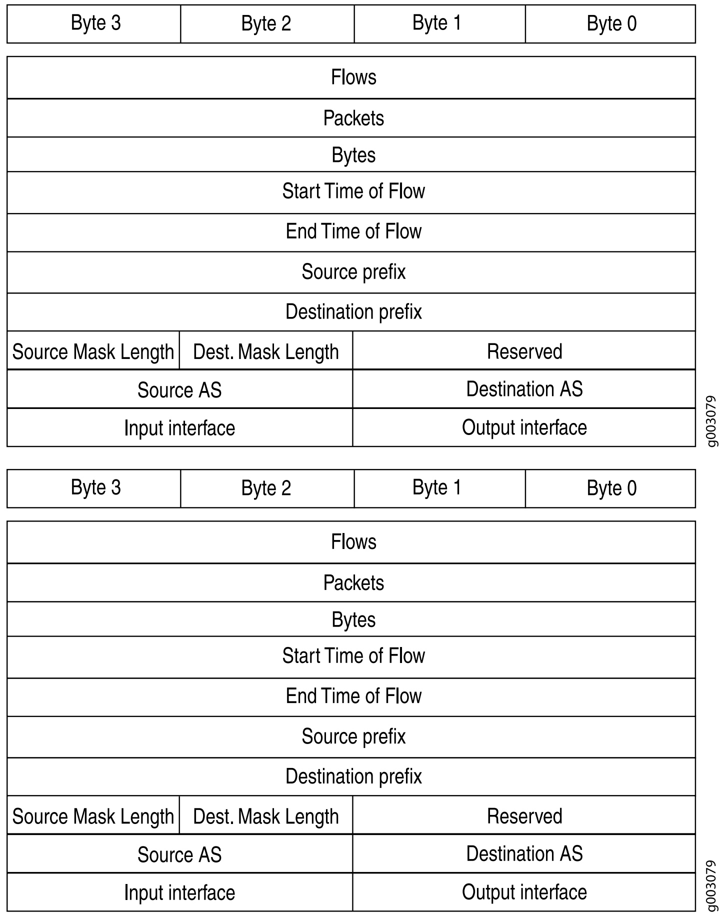


Table 8: Version 8 Prefix Aggregation Flow Entry Fields

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
Destination prefix	Destination IP address prefix
Source mask length	Source address network mask length

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

Field	Description
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

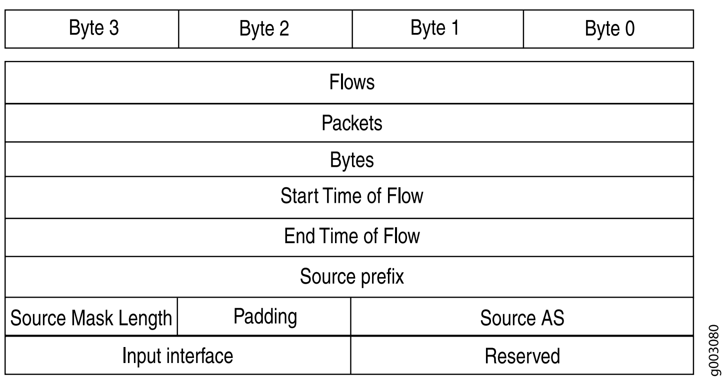


Table 9: Version 8 Source Prefix Aggregation Flow Entry Fields

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

Table 9: Version 8 Source Prefix Aggregation Flow Entry Fields *(continued)*

Field	Description
Source AS	AS number of the source address
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

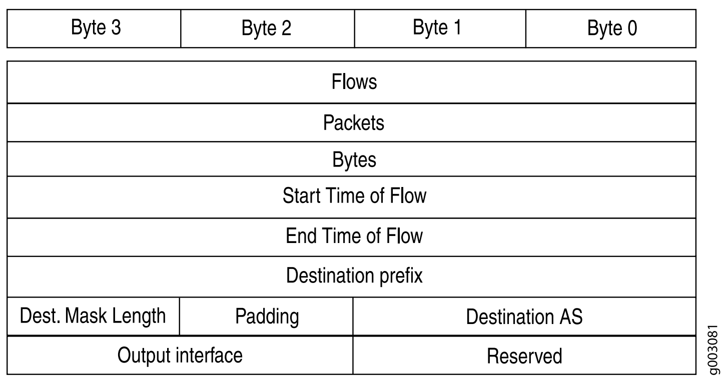


Table 10: Version 8 Destination Prefix Aggregation Flow Entry Fields

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
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**.

Version 9 Formats and Fields

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

- Table 11 on page 18
- Figure 9 on page 20
- Table 12 on page 20
- Figure 11 on page 23
- Table 12 on page 20
- Figure 12 on page 24
- Table 16 on page 24
- Figure 13 on page 25
- Table 17 on page 25

The Junos OS supports the version 9 template formats:

Table 11: 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 11: 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
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 11: 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 NOTE: Peer AS billing traffic is not supported for active flow monitoring version 9 configuration on PTX5000 routers tethered to CSE2000.

Figure 9: Version 9 Flow Header Format

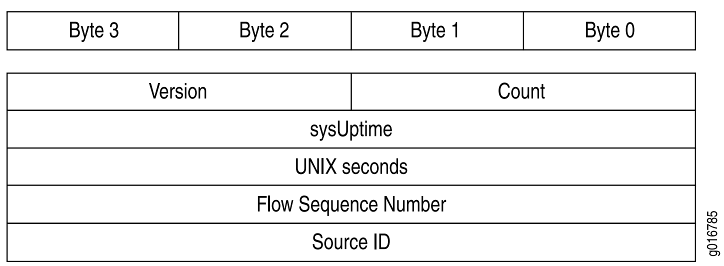


Table 12: Version 9 Flow Header Fields

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.

Table 12: Version 9 Flow Header Fields *(continued)*

Field	Description
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

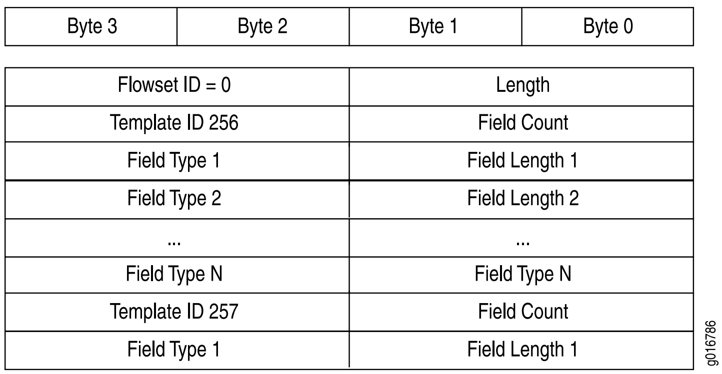


Table 13: Version 9 Template FlowSet Fields

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 14 on page 22.
Field Length	Length, in bytes, of the corresponding field type.

Table 14: 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.
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.

Table 14: Field Type Definitions Supported in Junos OS *(continued)*

Field Type	Description
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. 0x01 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.
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

Byte 3	Byte 2	Byte 1	Byte 0

Flowset ID = Template ID	Length
Record 1 - Field Value 1	Record 1 - Field Value 2
Record 1 - Field Value 3	...
Record 2 - Field Value 1	Record 2 - Field Value 2
Record 2 - Field Value 3	Record 2 - Field Value 2
Record 3 - Field Value 1	...
...	Padding

g016787

Table 15: 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.
Length	FlowSet length. Data FlowSets are fixed in length.

Table 15: Version 9 Data FlowSet Format *(continued)*

Field	Description
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

Byte 3	Byte 2	Byte 1	Byte 0

Byte 3–2	Byte 1–0
Flowset ID = 1	Length
Template ID	Option Scope Length
Option Length	Scope 1 Field Type
Scope 1 Field Length	...
Scope N Field Length	Option 1 Field Type
Option 1 Field Length	...
Option M Field Length	Padding

g016788

Table 16: 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.
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

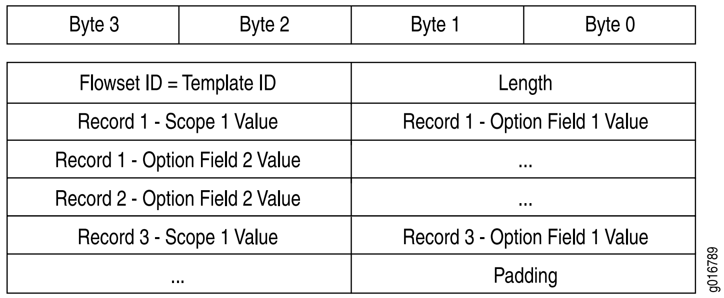


Table 17: Active Flow Monitoring Version 9 Options Data Record Format

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.

PART 2

Passive Flow Monitoring

CHAPTER 3

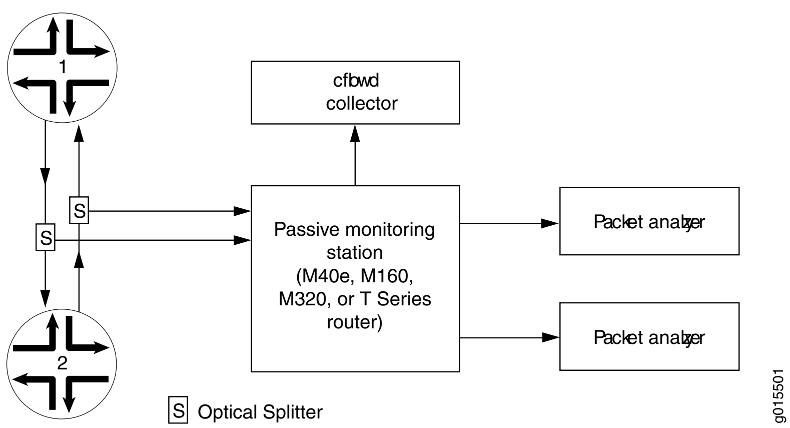
Understanding Passive Flow Monitoring

Passive Flow Monitoring Overview

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

The M40e, M160, M320, MX Series, or T 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 14 on page 29 shows a typical topology for the passive flow monitoring application.

Figure 14: 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

- Flow Monitoring Overview on page 3
- Active Flow Monitoring Overview on page 91

CHAPTER 4

System Requirements for Passive Flow Monitoring

Passive Flow Monitoring System Requirements

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

- Junos OS Release 9.2 or later for passive flow monitoring support for IQ2 interfaces only on M120, M320, T320, T640, T1600 and MX-series routers.
- Junos OS Release 8.5 or later for passive flow monitoring support on the MX Series MultiServices routers
- Junos OS Release 8.4 or later for passive flow monitoring support on the MultiServices 400 PIC (Type 2)
- Junos OS Release 7.6 or later to clear error and flow statistics with the **clear passive-monitoring statistics** command
- Junos OS Release 7.5 or later for support of the dynamic flow capture (DFC) Management Information Base (MIB)
- Junos OS Release 7.4 or later for dynamic flow capture on Monitoring Services III PICs installed in T Series and M320 routers, and port mirroring of IPv6 packets
- Junos OS Release 7.3 or later for passive flow monitoring on selected Ethernet-based interfaces and filter-based forwarding on output interfaces
- Junos OS Release 7.1 or later for passive flow monitoring and flow collection services on Monitoring Services II PICs installed in T Series and M320 routers
- Junos OS Release 6.4 or later for support of the next-hop IP address field in flow monitoring version 5 records
- Junos OS Release 6.2 or later for ATM2 intelligent queuing (IQ) interface passive monitoring, flow collection services, and MPLS label stripping
- Junos OS Release 6.1 or later for MPLS passive monitoring
- Junos OS Release 6.0 or later for the Monitoring Services II PIC

- Junos OS Release 5.7 or later for the automatic insertion of autonomous system (AS) numbers and SNMP index values for interfaces into flow records
- Junos OS Release 5.4 or later for the Monitoring Services PIC
- M40e, M160, M320, MX Series, or T Series router with an Internet Processor II ASIC or later
- 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:
  - SONET/SDH PIC—OC3, OC12, or OC48
  - ATM2 IQ PIC—OC3 or OC12
  - 4-port Fast Ethernet PIC
  - 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 on page 95
- Active Flow Monitoring PIC Specifications on page 96

Passive Flow Monitoring Hardware and Software Considerations

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 SONET/SDH interfaces (OC3, OC12, or OC48), ATM2 IQ interfaces (OC3 or OC12), 4-port Fast Ethernet interfaces, 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 SONET/SDH, ATM2 IQ, or Ethernet-based receive ports, one for each direction of flow. In Figure 14 on page 29, 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.

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 SONET/SDH interfaces is 4474 bytes; for Gigabit Ethernet and Fast Ethernet interfaces, it 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 or Fast 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.

CHAPTER 5

Configuring Passive Flow Monitoring

- Configuring a Monitoring Group to Send Traffic to the Flow Server on page 69
- Configuring Policy Options on page 70
- Option: Stripping MPLS Labels on ATM, Ethernet-Based, and SONET/SDH Interfaces on page 71
- Using a Flow Collector Interface to Process and Export Multiple Flow Records on page 72
- Example: Flow Collector Interface Configuration on page 78

Configuring Passive Flow Monitoring

Table 18 on page 36 shows which Juniper Networks PICs and routers support passive flow monitoring. The PICs receive passively monitored network traffic from an input interface (SONET/SDH, ATM2 IQ, Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet), convert the received packets into flow records, and export them to a flow server for further analysis.

Table 18: Passive Flow Monitoring PIC Support

PIC Type	M40e	M160	T Series/ M320
Monitoring Services PIC	Yes	Yes	No
Monitoring Services II PIC	Yes	Yes	Yes
Monitoring Services III PIC	Yes	Yes	Yes
MultiServices 400 PIC (Type 2)	Yes	No	Yes

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.

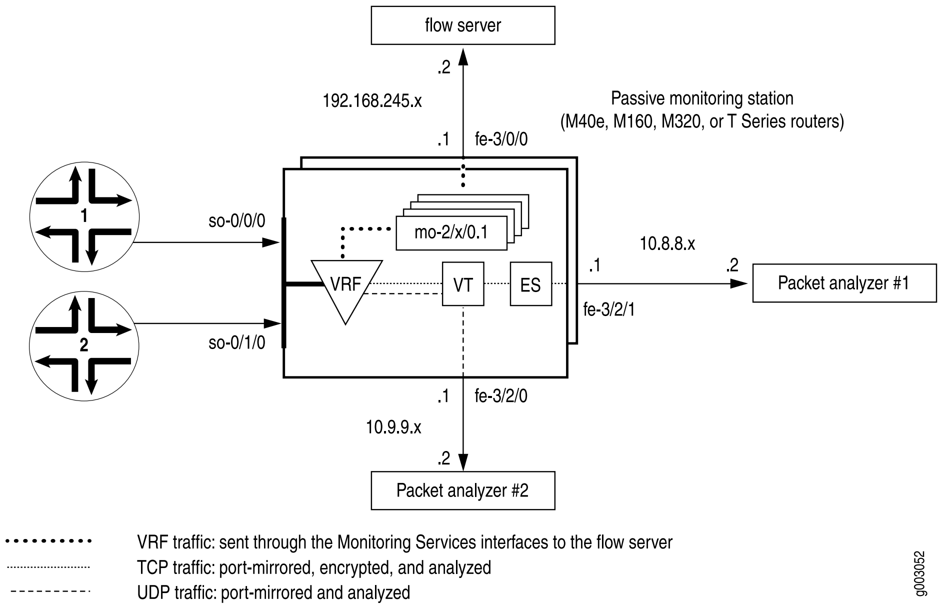
However, there are several options you can use that add complexity to passive flow monitoring. For example, you can configure the router to direct traffic into a routing instance and deliver the traffic into a monitoring group. You can also use port mirroring and filter-based forwarding to copy and redirect 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

- Copying and Redirecting Traffic with Port Mirroring and Filter-Based Forwarding on page 58
- Using a Flow Collector Interface to Process and Export Multiple Flow Records on page 72
- Using a Dynamic Flow Capture Interface to Monitor Traffic On Demand on page 50
- Passive Flow Monitoring Hardware and Software Considerations on page 32

Example: Passive Flow Monitoring Configuration

Figure 15: Passive Flow Monitoring—Topology Diagram



In Figure 15 on page 37, 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 19: 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.
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 20: 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.

Table 20: Output Fields for the show passive-monitoring flow Command *(continued)*

Field	Explanation
Flows inactive timed out	Total number of flows that are exported because of inactivity.
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 21: 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.

Table 21: Output Fields for the show passive-monitoring memory Command *(continued)*

Field	Explanation
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 22: 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).

Table 22: Output Fields for the show passive-monitoring status Command *(continued)*

Field	Explanation
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.
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 23: 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%
```

Using a Dynamic Flow Capture Interface to Monitor Traffic On Demand

Dynamic flow capture enables you to capture packet flows based on filtering criteria that you specify in real time. Unlike traditional flow monitoring that requires filtering criteria to be established before operation, dynamic flow capture uses an on demand control protocol that allows you to modify the filtering criteria as network conditions change.

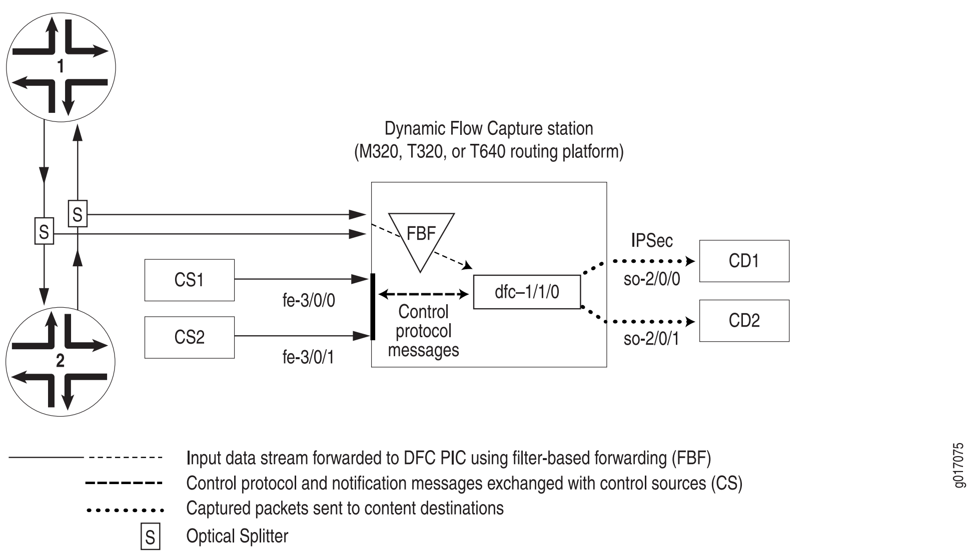
The dynamic flow capture architecture consists of one or more *control sources* that send Dynamic Tasking Control Protocol (DTCP) requests to a *monitoring station*. The requests contain filtering criteria that specify which incoming traffic should be monitored, and the monitoring station forwards any packets that match the filter criteria to a set of one or more *content destinations*.

- Control source—A client that wants to monitor electronic data or voice transfer over the network. The control source sends filter requests to the Juniper Networks router using DTCP. The control source is identified by a unique identifier and an optional list of IP addresses.
- Monitoring station—A Juniper Networks T Series or M320 router configured with one or more Monitoring Services III PICs which support dynamic flow capture processing. The monitoring station 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 station. Typically the matched packets are sent using an IPSec tunnel from the monitoring station to another router connected to the content destination. The content destination and the control source can be located on the same host.

NOTE: **The DFC PIC forwards the entire packet content to the content destination, rather than just a content record.**

Figure 16 on page 51 shows a sample topology that contains control sources, a monitoring station, and content destinations.

Figure 16: Dynamic Flow Capture Topology



To configure dynamic flow capture, perform the following tasks:

- Configuring the Dynamic Flow Capture Group on page 52
- Configuring the Content Destination for Dynamic Flow Capture on page 52

Configuring the Dynamic Flow Capture Group

A dynamic flow capture group defines a profile of dynamic flow capture configuration information. 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:

```
[edit services dynamic-flow-capture]
capture-group client-name {
  content-destination identifier {
    address address;
    ttl hops;
  }
  control-source identifier {
    allowed-destinations [ destination ];
    no-syslog;
    notification-targets [ address address port  port-number ];
    service-port port-number;
    shared-key value;
    source-addresses [ address ];
  }
  input-packet-rate-threshold rate;
  interfaces interface-name;
  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 for Dynamic Flow Capture

You must specify a destination for the packets that match dynamic flow capture filter criteria. To configure, include the **content-destination** statement at the [**edit services dynamic-flow-capture capture-group *client-name***] hierarchy level:

```
[edit services dynamic-flow-capture capture-group client-name]
content-destination  identifier {
  address address;
  ttl hops;
}
```


Assign the **content-destination** a unique ***identifier***. In addition, you must specify its IP address, and you can optionally set a time-to-live (TTL) value for the IP-IP header:

- **address**—The dynamic flow capture 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**—By default, the TTL value is 255, with a range from 0 through 255.

Configuring the Control Source for Dynamic Flow Capture

You configure information about the control source, including allowed source addresses, destinations, and authentication key values. To configure the control source information, include the **control-source** statement at the [**edit services dynamic-flow-capture**] hierarchy level:

```
[edit services dynamic-flow-capture capture-group client-name]
control-source identifier {
  allowed-destinations [ destination-identifier ];
  no-syslog;
  notification-targets [ address address port port-number ];
  service-port port-number;
  shared-key value;
  source-addresses [ address ];
}
```

Assign the **control-source** statement with 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 matched data to be sent in its control protocol requests. If you do not specify any content destinations, all available destinations are allowed.
- **notification-targets**—One or more destinations to which the dynamic flow capture interface can log information about control protocol-related events and other events such as PIC startup 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 dynamic flow capture interfaces.
- **shared-key**—A 20-byte authentication key value shared between the control source and the dynamic flow capture monitoring station.
- **source-addresses**—One or more allowed IP addresses from which the control source can send control protocol requests to the dynamic flow capture monitoring station. These are /32 addresses.

Configuring the Dynamic Flow Capture Interface

You specify the interface that interacts with the control sources configured in the same dynamic flow 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 dynamic flow capture interface, include the **interfaces** statement at the [**edit services dynamic-flow-capture**] hierarchy level:

```
[edit services dynamic-flow-capture capture-group client-name]
interfaces interface-name;
```

You specify dynamic flow capture interfaces using the **dfc-** identifier at the **[edit interfaces]** hierarchy level. Three logical units are required on each dynamic flow capture 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 dynamic flow capture interface:

```
[edit interfaces dfc-0/0/0]
unit 0 {
  family inet {
    address 10.1.0.0/32 { # Address of the Routing Engine for the DFC PIC.
      destination 10.36.100.1; # Address of DFC PIC; used by the
        # control source to correspond with the monitoring station.
    }
  }
}
unit 1 { # Receives data packets on this logical interface.
  family inet;
}
unit 2 { # Sends copies of matched packets from this logical interface.
  family inet;
}
```

In addition, you must configure the dynamic flow capture application to run on the DFC PIC in the correct chassis location. The following example shows this configuration at the **[edit chassis]** hierarchy level:

```
[edit chassis]
fpc 0 {
  pic 0 {
    monitoring-services application dynamic-flow-capture;
  }
}
```

For more information on configuring chassis properties, see the *Junos System Basics Configuration Guide*.

Option: Configuring System Logging for Dynamic Flow Capture

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:

```
[edit syslog]
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:

```
[edit services dynamic-flow-capture capture-group client-name control-source identifier]
no-syslog;
```

Option: Configuring Thresholds for Recording System Log Messages for Dynamic Flow Capture Interfaces

You can optionally specify threshold values for situations in which warning messages will be recorded in the system log:

- Input packet rate to the dynamic flow capture interfaces
- Memory usage on the dynamic flow capture interfaces

To configure, include the **input-packet-rate-threshold** or **pic memory-threshold** statements at the [**edit services dynamic-flow-capture capture-group** ***client-name***] hierarchy level:

```
[edit services dynamic-flow-capture capture-group  client-name]
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.

Option: Monitoring Dynamic Flow Capture Using SNMP

In Junos OS Release 7.5 and later, 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. For more information, see the *Junos Network Management Configuration Guide*.

Example: Dynamic Flow Capture Configuration

The following example shows a complete dynamic flow capture configuration. On Router 1, configure the dynamic flow capture interface, the interfaces that connect to the control source and content destination, and the interface that receives passively monitored traffic. Then, configure the capture group and specify your control source and content destination requirements. Next, configure filter-based forwarding (FBF) to send monitored traffic to logical unit 1 of the dynamic flow capture interface. Finally, configure a firewall filter and routing table groups to complete the configuration.

```
[edit]
interfaces {
  dfc-0/0/0 { # DFC PIC that processes requests from the control source.
    unit 0 {
      family inet {
        address 192.0.2.0/32 { # Address of the Routing Engine for the DFC PIC.
          destination 10.36.100.1; # Address of DFC PIC; used by
        } # the control source to communicate with the monitoring station.
      }
    }
    unit 1 { # This logical interface receives data packets.
      family inet;
    }
    unit 2 {  # This logical interface sends out copies of matched packets.
      family inet;
    }
  }
  fe-4/1/2 { # Interface that receives filtering requests from cs1.
    unit 0 {
      family inet {
        address 10.36.41.2/30;
      }
    }
  }
  ge-7/0/0 {  # Interface that sends monitored packets to cd1.
    unit 0 {
      family inet {
        address 10.36.70.1/30;
      }
    }
  }
  so-1/2/0 {  # Interface that receives traffic to be monitored.
    encapsulation ppp;
    unit 0 {
      passive-monitor-mode; # Enables this interface to be passively monitored.
      family inet {
        filter {
          input catch;
        }
      }
    }
  }
}
services {
  dynamic-flow-capture {
    capture-group g1 {
      interfaces dfc-0/0/0; # Specifies which interface to use for DFC processing.
      input-packet-rate-threshold 90k; # Traffic threshold for system log messages.
      pic-memory-threshold percentage 80; # Memory threshold for log messages.
      control-source cs1 { # Specifies addresses and ports for the control source.
        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 { # Specifies content destination addresses and TTL.
            address 10.36.70.2;
            ttl 244;
         }
      }
   }
}
firewall {
   filter catch { # Places monitored traffic into the filter-based forwarding instance.
      interface-specific;
      term def {
         then {
            count counter;
            routing-instance fbf_inst;
         }
      }
   }
}
routing-instances {
   fbf_inst { # Sends matching traffic to the DFC PIC for processing.
      instance-type forwarding;
      routing-options {
         static {
            route 0.0.0.0/0 next-hop dfc-0/0/0.1;
         }
      }
   }
}
routing-options {
   interface-routes {
      rib-group inet common;
   }
   rib-groups {
      common { # Shares routes between the instance and the main routing table.
         import-rib [ inet.0 fbf_inst.inet.0 ];
      }
   }
   forwarding-table {
      export pplb;
   }
}
```

Verifying Your Work

To verify that your dynamic flow capture configuration is operating correctly, issue the following command:

show services dynamic-flow-capture capture-group ***group-name*** control-source
source-identifier ***source-id*** (detail)

The following section shows the output of this command when used with the configuration example.

Router 1

user@router1> **show services dynamic-flow-capture control-source capture-group g1 source-identifier cs2 detail**

```
Capture group: g1, Control source: cs2
Criteria added: 1, Criteria add failed: 0
Active criteria: 2
Static criteria: 0, Dynamic criteria: 2
Control protocol requests: 3
      Add        Delete          List       Refresh         No-op
Requests           1              0             1             0             1
Failed             0              0             0             0             0

Add request rate: 0
Add request peak rate: 1
Bandwidth across all criteria: 0
Total notifications: 0
Restart: 0, Rollover: 0, No-op: 0, Timeout: 0, Congestion: 0, Congestion delete: 0,
Dups dropped: 0
Criteria deleted: 0
Timeout idle: 0, Timeout total: 0, Packets: 0, Bytes: 0
Sequence number: 242
```

To clear dynamic flow capture criteria belonging to a particular control source, issue the **clear services dynamic-flow-capture** command. For more information on other dynamic flow capture-related operational mode commands, see the *Junos System Basics and Services Command Reference*.

Copying and Redirecting Traffic with Port Mirroring and Filter-Based Forwarding

This section discusses additional techniques you can use with the passive flow monitoring application:

- In addition to flow analysis, you can analyze a copy of the original traffic with a single packet analyzer. To implement this technique, divert traffic with a filter-based forwarding routing instance and send the monitored traffic through a physical interface to the packet analyzer.
- You can cluster the traffic into different groups and redirect this traffic to multiple packet analyzers. For example, you can break traffic flows into TCP groups and UDP groups and send these groups of packets to different analyzers. To accomplish this, you use port mirroring and send a copy of the original traffic to a Tunnel PIC. Then you can apply a firewall filter, split the traffic into your desired groups, and send these groups toward different exit interfaces leading to the packet analyzers. This technique provides maximum flexibility for traffic analysis.
- For secure transmission of the copied or grouped traffic, you can encrypt the diverted traffic with an ES PIC and send this traffic to a packet analyzer over an IP Security (IPSec) tunnel.

To implement the filter-based forwarding enhancement methods, see the following sections:

Specifying Port Mirroring Input and Output

This step works in conjunction with the action specified by the **port-mirror** statement configured at the **[edit firewall family (inet | inet6) filter *filter-name* term *term-name* then]** hierarchy level. At this point, you select input and output statements to determine where the copies of the IPv4 or IPv6 packets are sent. To configure, include the **input** and **output** statements at the **[edit forwarding-options port-mirroring family *family-name*]** hierarchy level. The traffic to be monitored is copied, port-mirrored, and sent to the packet analyzer for analysis. On M Series routers, you can port-mirror either IPv4 or IPv6 packets at one time. On M120, M320, and T Series routers, you can port-mirror both IPv4 and IPv6 packets simultaneously.

NOTE: On an M320 or T Series router using an Adaptive Services (AS) II PIC or a MultiServies PIC, corrupted IP packets might be sent to the port mirror when traffic passes through an IPSec tunnel. The inbound IP traffic passes through the IPSec tunnel and the sp interface is decoded and forwarded to the port mirror correctly, but the return outbound traffic is corrupted and unreadable through the router configured with the port mirror.

The port-mirrored copy of the traffic can travel only to a single next hop. As a result, only one type of analysis can be performed if the packets are sent to a packet analyzer through a physical next hop. If more than one type of analysis is desired, a tunnel interface must be used as the next hop for port mirroring. When the mirrored copy of the traffic arrives at the virtual tunnel interface, it can be filtered, split into groups, and redirected to multiple exit interfaces and packet analyzers.

For your input requirements, include the **rate** and **run-length** statements at the **[edit forwarding-options port-mirroring family *family-name* input]** hierarchy level. For your output requirements, specify the target interface with the **interface** statement at the **[edit forwarding-options port-mirroring family *family-name* output]** hierarchy level.

By default, a filter cannot be applied to an interface where port-mirrored traffic is received. To allow the tunnel services interface to be used as a filtered next hop, include the **no-filter-check** statement at the **[edit forwarding-options port-mirroring family *family-name* output]** hierarchy level.

```
[edit]
forwarding-options {
   port-mirroring {
      family (inet | inet6) {
         input {
            rate 1;
            run-length 5;
         }
         output {
            interface vt-0/2/0.0;
            no-filter-check;
         }
      }
   }
}
```

NOTE: Before Junos OS Release 7.4, you could configure the **input** and **output** statements at the **[edit forwarding-options port-mirroring]** hierarchy level. However, this older syntax has been revised to extend port-mirroring support to IPv6 packets. If you have a configuration that contains the older syntax, we recommend that you update your configuration to the new syntax listed above.

Creating a Firewall Filter to Split the Port-Mirrored Traffic into Different Instances

If you need to split the copy of the monitored traffic into separate groups and send these filtered packets to different analyzers, devise a firewall filter that selects some traffic for sampling and some traffic for discarding. In this case, UDP traffic is sent into one routing instance, TCP traffic is diverted into a second routing instance, and all other traffic is discarded. In a later step, you will define the filter-based forwarding routing instances specified in the **then** statements shown in this filter.

```
[edit]
firewall {
   family inet {
      filter tunnel-interface-filter {
         term tcp {
            from {
               protocol tcp;
            }
            then {
               count tcp;
               routing-instance tcp-routing-table;
            }
         }
```

```
            term udp {
               from {
                  protocol udp;
               }
               then {
                  count udp;
                  routing-instance udp-routing-table;
               }
            }
            term rest {
               then {
                  count rest;
                  discard;
               }
            }
         }
      }
   }
```

Applying the Firewall Filter to a Tunnel PIC Interface

Once the firewall filter is defined, apply it as an input filter on a tunnel interface. This is required if the firewall filter defines two or more types of traffic or export interfaces. However, if the firewall filter only specifies one type of traffic and one export interface, you can apply the filter directly to the export interface.

```
[edit]
interfaces {
   vt-0/2/0 {
      unit 0 {
         family inet {
            filter {
               input tunnel-interface-filter;
            }
         }
      }
   }
}
```

Using Filter-Based Forwarding to Export Monitored Traffic to Multiple Destinations

The firewall filter called **tunnel-interface-filter** that you made earlier sends UDP traffic into one filter-based forwarding routing instance called **udp-routing-table**, sends TCP traffic into a second filter-based forwarding routing instance called **tcp-routing-table**, and discards all other packets. Here you will configure the filter-based forwarding instances.

Configure an export interface for each of your routing instances by including a static next hop. To configure, include the **route** statement at the **[edit routing-instances *instance-name* routing-options static]** hierarchy level and specify a next-hop address or interface.

```
[edit]
routing-instances {
```

```
    tcp-routing-table {
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop es-3/1/0.0;
            }
        }
    }
    udp-routing-table {
        instance-type forwarding;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 10.9.1.2;
            }
        }
    }
}
```

Configuring a Routing Table Group 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;
        }
    }
}
```

Option: Using an ES PIC to Send Traffic to a Packet Analyzer

You can send some or all of the traffic securely to the packet analyzer using IPSec and an 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 Feature 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";
        }
    }
}
```

Option: Applying a Firewall Filter to an Output Interface

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 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 to Select Traffic to Monitor on page 64
- *Configuring Input Interfaces, Monitoring Services Interfaces, and Export Interfaces*
- Establishing a VRF Instance for the Monitored Traffic on page 68
- *Configuring a Monitoring Group to Send Traffic to the Flow Server*
- *Configuring Policy Options*
- *Option: Stripping MPLS Labels on ATM, Ethernet-Based, and SONET/SDH Interfaces*

Specifying a Firewall Filter 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

After creating the input filter, you need to configure the interfaces where traffic will enter the router. To enable passive flow monitoring for SONET/SDH input interfaces, include the **passive-monitor-mode** statement at the **[edit interfaces so-*fpc/pic/port* unit *unit-number*]** hierarchy level. This mode disables the router from participating in the network as an active device. On SONET/SDH interfaces, passive monitor mode suppresses SONET keepalives.

For ATM2 IQ interfaces, passive monitor mode suppresses the sending and receiving of ATM Operations, Administration, and Maintenance (OAM) and Integrated Local Management Interface (ILMI) control messages. To enable passive flow monitoring for ATM2 IQ input interfaces, include the **passive-monitor-mode** statement at the **[edit interfaces at-*fpc/pic/port*]** hierarchy level. ATM passive monitoring supports the following interface encapsulation types: Cisco-compatible ATM Network Layer Protocol ID (NLPID) (**atm-cisco-nlpid**), ATM NLPID (**atm-nlpid**), ATM Point-to-Point Protocol (PPP) over ATM Adaptation Layer 5 (AAL5)/ logical link control (LLC) (**atm-ppp-llc**), ATM PPP over raw AAL5 (**atm-ppp-vc-mux**), ATM LLC/ subnetwork attachment point (SNAP) (**atm-snap**), and ATM virtual circuit (VC) multiplexing (**atm-vc-mux**).

Ethernet-based interfaces support both per-port passive monitoring and per-VLAN passive monitoring. For Fast Ethernet interfaces, include the **passive-monitor-mode** statement at the **[edit interfaces fe-*fpc/pic/port*]** hierarchy level. 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 (fe | 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]** or **[edit interfaces fe-*fpc/pic/port* fastether-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 {
  so-0/0/0 {
    description "SONET/SDH input interface";
    encapsulation ppp;
    unit 0 {
      passive-monitor-mode;
      family inet {
        filter {
          input  input-monitoring-filter;
        }
      }
    }
  }
  at-1/0/0 {
    description "ATM2 IQ input interface";
    passive-monitor-mode;
    atm-options {
      pic-type atm2;
      vpi 0 {
        maximum-vcs 255;
      }
    }
    unit 0 {
      encapsulation atm-snap;
      vci 0.100;
      family inet {
        filter {
          input input-monitoring-filter;
        }
```



```
            }
         }
      }
      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" on page 61.

```
[edit]
interfaces
fe-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 for the 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 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.

NOTE: Because routing instances determine the input interface, the **input** statement at the **[edit forwarding-options monitoring *group-name* family inet]** hierarchy level has been removed in Junos OS Release 6.0 and later. If you have a configuration that contains this old statement, we recommend that you update your configuration and remove the statement.

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" on page 58.

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 page 65.

```
[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

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;
    }
  }
}
```

Option: Stripping MPLS Labels on ATM, Ethernet-Based, and SONET/SDH 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 ATM2 IQ, Ethernet-based, or SONET/SDH interface, include the **pop-all-labels** statement at the **[edit interfaces *interface-name-fpc/pic/port* (atm | fastether | gigether | sonet)-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* (atm | fastether | gigether | sonet)-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 {
  at-/fpc/pic/port {
    atm-options {
```

```
            mpls {
                pop-all-labels {
                    required-depth 1;
                }
            }
        }
    }
    (fe | ge)-fpc/pic/port {
        (fastether | gigether)-options {
            mpls {
                pop-all-labels {
                    required-depth [1 2];
                }
            }
        }
    }
    so-fpc/pic/port {
        sonet-options {
            mpls {
                pop-all-labels {
                    required-depth 2;
                }
            }
        }
    }
}
```

Using a 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 24 on page 73.

Table 24: 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**

Table 24: Name Format Macros *(continued)*

Field	Expansion
{*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**
{*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, see the *Junos Network Management Configuration Guide* or view the enterprise-specific Juniper Networks MIBs at **http://www.juniper.net/techpubs/software/junos/mibs.html**.

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: Flow Collector Interface Configuration" on page 78.

```
[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;
        }
    }
    SONET/SDH, ATM2 IQ, or Ethernet-based-interface-fpc/pic/port {
        description " input_interface";
        encapsulation encapsulation-type;
        passive-monitor-mode; # Apply to the logical interface for SONET/SDH
    }
}
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: Flow Collector Interface Configuration

Figure 17: Flow Collector Interface Topology Diagram

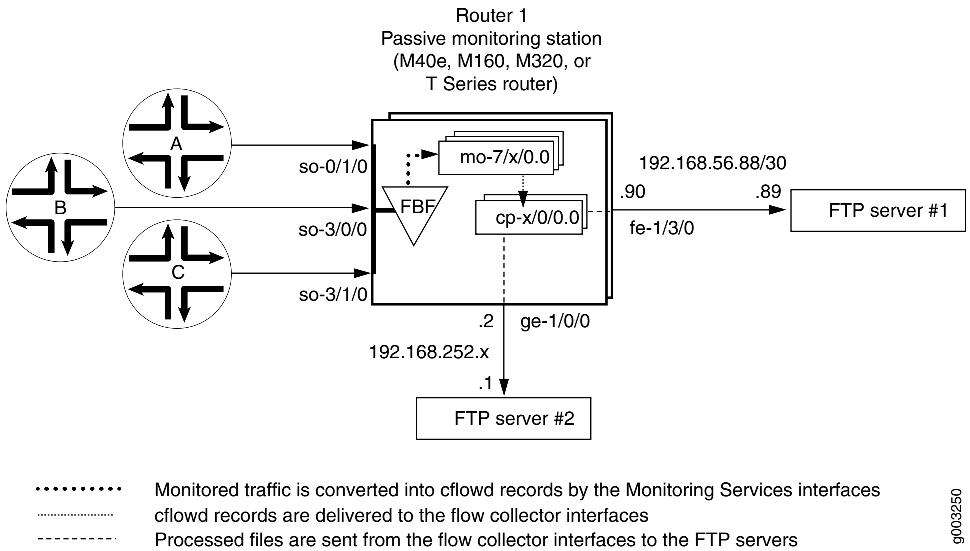


Figure 17 on page 78 shows the path traveled by monitored traffic as it passes through the router. Packets arrive at input interfaces **so-0/1/0**, **so-3/0/0**, and **so-3/1/0**. The raw packets are directed into a filter-based forwarding routing instance and processed into flow records by the monitoring 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 25 on page 86 explains the various fields available in the transfer log.

Table 25: 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 26 on page 87 explains the various fields available in the flow collector interface file.

Table 26: 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

Table 26: Flow Collector Interface File Fields in Order of Appearance *(continued)*

Field	Explanation
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
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
```


PART 3

Active Flow Monitoring

CHAPTER 6

Understanding Active Flow Monitoring

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 on page 3
- Passive Flow Monitoring Overview on page 29

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:

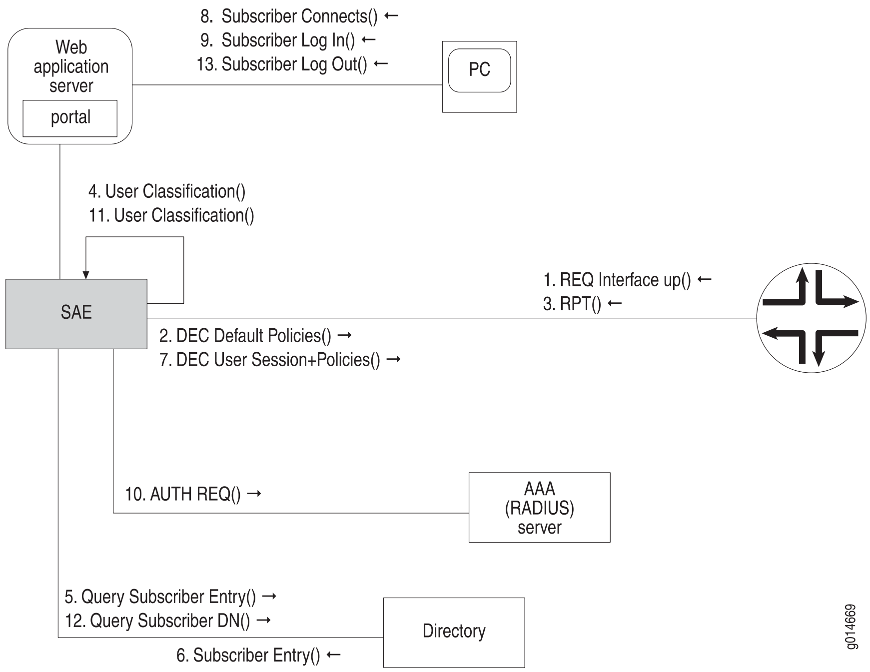
- Tracking what kind of traffic is entering or exiting an ISP or corporate network.
- Tracking traffic flows between BGP autonomous systems.
- Tracking traffic flows between enterprise network regions.
- Taking a snapshot of the existing quality-of-service (QoS) policy results prior to making changes 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 18: Active Flow Monitoring



Related Documentation

- Flow Monitoring Overview on page 3
- Active Flow Monitoring Overview on page 91

CHAPTER 7

System Requirements for Active Flow Monitoring

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 on page 31
- Active Flow Monitoring PIC Specifications on page 96

Active Flow Monitoring PIC Specifications

For Monitoring Services PIC specifications, see Table 27 on page 96 and Table 28 on page 97. For Adaptive Services PIC specifications, see Table 29 on page 97. For MultiServices PIC specifications, see Table 30 on page 97 and Table 31 on page 98.

Table 27: 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 28: Monitoring Services II 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.
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 29: 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.
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 30: MultiServices 100 PIC

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

Table 30: MultiServices 100 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.

Table 31: 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 32: MultiServices 500 PIC

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

Table 32: 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 on page 31
- Active Flow Monitoring System Requirements on page 95

CHAPTER 8

Configuring Active Flow Monitoring

- Example: Multiple Port Mirroring with Next-Hop Groups Configuration on page 150
- Example: Sampling Configuration on page 154
- Example: Sampling Instance Configuration on page 158
- Example: Sampling and Discard Accounting Configuration on page 164

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 33 on page 102 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 33: 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
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

Table 33: 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
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 to Select Traffic for Active Flow Monitoring on page 124
- Configuring the Interfaces That Will Be Actively Monitored on page 120
- Enabling the Monitoring Services, Adaptive Services, or Multiservices Interfaces and the Export Interface on page 125
- Collecting Flow Records on page 121
- Option: Configuring Port Mirroring on page 148
- Option: Configuring Port Mirroring with Filter-Based Forwarding and a Monitoring Group on page 148
- Option: Sending Port-Mirrored Traffic to Multiple Export Interfaces by Using Next-Hop Groups on page 149
- Option: Using the Flow-Tap Application to Send Packets to a Mediation Device on page 107

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*

Option: Using the Flow-Tap Application to Send Packets to a Mediation Device

Dynamic flow capture enables you to capture passively monitored packet flows on the basis of dynamic filtering criteria, using Dynamic Tasking Control Protocol (DTCP) requests. The flow-tap application extends the use of DTCP to intercept IPv4 packets in an active flow monitoring station and send a copy of packets that match filter criteria to one or more content destinations. Flow-tap data can be used for lawful intercept purposes and provides flexible trend analysis for detection of new security threats. The flow-tap application is supported on M Series and T Series routers, except M160 routers and TX Matrix platforms.

NOTE: For information about dynamic flow capture, see "Using a Dynamic Flow Capture Interface to Monitor Traffic On Demand" on page 50. For information about DTCP, see Internet draft draft-cavuto-dtcp-01.txt at http://www.ietf.org/internet-drafts.

For detailed information about the flow-tap application, see the following sections:

- Flow-Tap Architecture on page 107
- Configuring the Flow-Tap Interface on page 109
- Configuring Flow-Tap Security Properties on page 110
- Flow-Tap Application Restrictions on page 110
- Example: Flow-Tap Configuration on page 111

Flow-Tap Architecture

The flow-tap architecture consists of one or more *mediation devices* that send requests to a Juniper Networks router 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 Juniper Networks router using the DTCP. The clients are not identified for security reasons, but have permissions defined by a set of special login classes.
- Monitoring platform—A Juniper Networks M Series or T Series router containing one or more Adaptive Services (AS) PICs, which are configured to support the flow-tap 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.

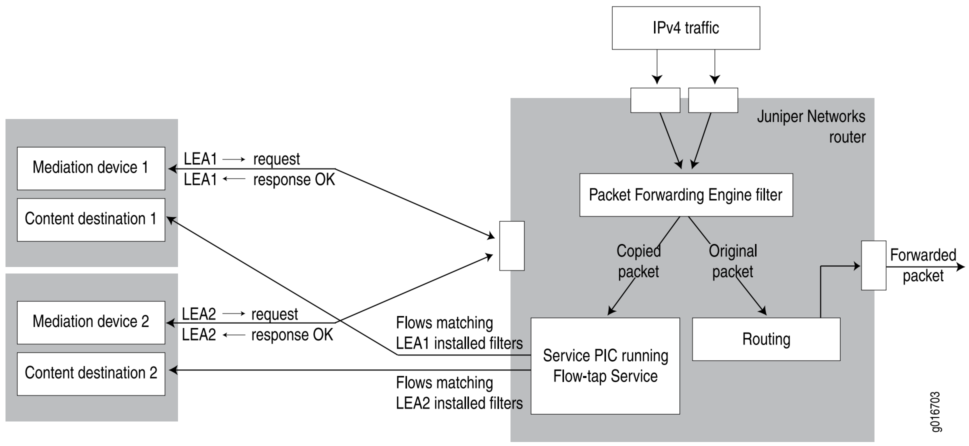
- Content destination—Recipient of the matched packets from the monitoring platform. Typically the matched packets are sent using an IP Security (IPSec) tunnel from the monitoring platform to another router connected to the content destination. The content destination and the mediation device can be 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 AS PIC that is configured for flow-tap service. The AS PIC 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.;
      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 19 on page 109 shows a sample topology that uses two mediation devices and two content destinations.

Figure 19: Flow-Tap Topology Diagram



Related Documentation

- Configuring the Flow-Tap Interface on page 109
- Configuring Flow-Tap Security Properties on page 110
- Flow-Tap Application Restrictions on page 110
- Example: Flow-Tap Configuration on page 111

Configuring the Flow-Tap Interface

To configure an AS PIC interface for the flow-tap 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 flow-tap service, and use any logical unit on the PIC.

NOTE: You cannot configure dynamic flow capture and flow-tap 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

- Flow-Tap Architecture on page 107
- Configuring Flow-Tap Security Properties on page 110
- Flow-Tap Application Restrictions on page 110
- Example: Flow-Tap Configuration on page 111

Configuring Flow-Tap Security Properties

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 flow-tap features are as follows:

- **flow-tap**—Can view flow-tap configuration.
- **flow-tap-control**—Can modify flow-tap 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"
```

For details on **[edit system]** and RADIUS configuration, see the *Junos System Basics Configuration Guide*.

Related Documentation

- Flow-Tap Architecture on page 107
- Configuring the Flow-Tap Interface on page 109
- Flow-Tap Application Restrictions on page 110
- Example: Flow-Tap Configuration on page 111

Flow-Tap Application Restrictions

The following restrictions apply to flow-tap services:

- You cannot configure dynamic flow capture and flow-tap services on the same router simultaneously.
- When the dynamic flow capture process or an AS PIC configured for flow-tap 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.

- Port mirroring might not work in conjunction with flow-tap processing.
- If the flow-tap application is configured, you cannot configure the filter action **then syslog** for any firewall filter running on the same platform.
- Running the flow-tap application over an IPsec tunnel on the same router can cause packet loops and is not supported.

Related Documentation

- Flow-Tap Architecture on page 107
- Configuring the Flow-Tap Interface on page 109
- Configuring Flow-Tap Security Properties on page 110
- Example: Flow-Tap Configuration on page 111

Example: Flow-Tap Configuration

The following example shows all the parts of a complete flow-tap configuration.

NOTE: The following example applies only to M Series and T Series routers, except M160 and TX Matrix routers. For MX Series routers, because the flow-tap application resides in the Packet Forwarding Engine rather than a service PIC or Dense Port Concentrator (DPC), the Packet Forwarding Engine must send the packet to a tunnel logical (vt-) interface to encapsulate the intercepted packet. In such a scenario, you need to allocate a tunnel interface and assign it to the dynamic flow capture process for FlowTapLite to use.

```
services {
  flow-tap {
    interface sp-1/2/0.100;
  }
}
interfaces {
  sp-1/2/0 {
    unit 100 {
      family inet;
    }
  }
}
system {
  services {
    flow-tap-dtcp {
      ssh {
        connection-limit 5;
        rate-limit 5;
      }
    }
  }
  login {
    class ft-class {
      permissions flow-tap-operation;
```

```
      }
      user ft-user1 {
         class ft-class;
         authentication {
            encrypted-password "xxxx";
         }
      }
   }
}
```

Related Documentation

- Flow-Tap Architecture on page 107
- Configuring the Flow-Tap Interface on page 109
- Configuring Flow-Tap Security Properties on page 110
- Flow-Tap Application Restrictions on page 110

Configuring FlowTapLite

A lighter version of the flow-tap application is available on MX Series routers and also on M320 routers with Enhanced III Flexible PIC Concentrators (FPCs). All of the functionality resides in the Packet Forwarding Engine rather than a service PIC or Dense Port Concentrator (DPC).

NOTE: **On M320 routers only, if the replacement of FPCs results in a mode change, you must restart the dynamic flow capture process manually by disabling and then re-enabling the CLI configuration.**

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 flow-tap 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;
}
```

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.

For more information about this configuration, see the *Junos OS Administration Library for Routing Devices*.

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 or DPC is available, you can use its tunnel interface for the same purpose.

NOTE: If you do not include the **family intet6** 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.

Related Documentation

- *Understanding Junos Packet Vision*
- *Configuring Junos Packet Vision*
- *Examples: Configuring Junos Packet Vision*

Example: IPv6 Support for FlowTapLite

This example describes how to configure IPv6 support for FlowTapLite on an M120 router with Enhanced III FPCs. The configuration of FlowTapLite is similar on an M320 router and an MX Series router with Enhanced III FPCs. However, because the MX Series routers do not support Tunnel Services PICs, you configure a DPC and the corresponding Packet Forwarding Engine to use tunneling services at the **[edit chassis]** hierarchy level.

With Junos OS Release 10.1, the FlowTapLite service supports lawful interception of IPv6 packets; previously only interception of IPv4 packets was supported. The intercepted packets are sent to a content destination, while the flow of original packets to the actual destination is unaffected.

A mediation device installs dynamic filters on the router (or server) by sending DTCP requests. These filters include the quintuple information (source address, destination address, source port, destination port, and protocol) about the intercepted flows and the details (IP addresses and port information) of the content destination.

Below is an example of such a filter:

```
ADD DTCP/0.8
Csource-ID: ftap
Cdest-ID: cd1
Source-Address: 2001:0DB8:ABCD:EF12:3456:78AB:ABC8:1235/112
Dest-Address: affe::1:1
Source-Port: 1234
Dest-Port: 2345
Protocol: *
Priority: 2
X-JTap-Input-Interface: ge-2/0/1
X-JTap-Cdest-Dest-Address: 192.0.2.5
X-JTap-Cdest-Dest-Port: 2300
X-JTap-Cdest-Source-Address: 198.51.100.9
X-JTap-Cdest-Source-Port: 65535
X-JTap-Cdest-TTL: 255
X-JTap-IP-Version: ipv6
Flags: STATIC
```

Following are descriptions of the parameters in the dynamic filter:

- **Csource-ID**—The username configured in the router at the [**edit system login user]** hierarchy level.
- **Cdest-ID**—The content destination identifier.
- **Source-Address**, **Dest-Address Source-Port**, **Dest-Port**, **Protocol**—Parameters that determine which packet flows need to be intercepted.
- **X-JTap-Input-Interface**—The interface through which the actual flows are coming into the router. Depending on the type of filters installed, the value in this field can include the following: **X-JTap-Output-Interface** to install output interface filters; **X-JTap-VRF-NAME** to install VRF filters; and to install global filters, no parameters are specified.

- **X-JTap-Cdest-Dest**—All parameters that start with this string specify different parameters associated with the content destination.
- **X-JTap-IP-Version**–Differentiates between IPv6 and IPv4 filters.

From the Packet Forwarding Engine console, you can verify that the filters are installed and working correctly.

This example describes how to configure IPv6 support for FlowTapLite on an M120 router:

- Requirements on page 115
- Overview and Topology on page 115
- Configuration on page 116
- Verification on page 118

Requirements

This example uses the following hardware and software components:

- Junos OS Release 10.1 or later
- M120 router with a tunnel (**vt**) interface

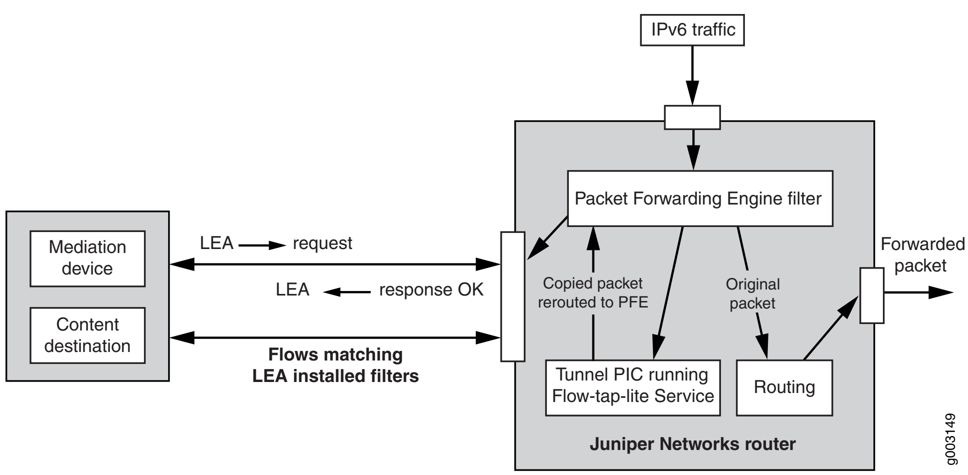
Before you configure IPv6 FlowTapLite on your router, be sure you have:

- A tunnel PIC that is up
- A connection from the router to the mediation device and the content destination
- Traffic flow to and from the router

Overview and Topology

Figure 20 on page 115 shows the FlowTapLite configuration for one M120 router to lawfully intercept packets.

Figure 20: FlowTapLite Topology



In this example, the IPv6 packets enter the Packet Forwarding Engine and, depending on the filters installed, a new flow is created for the intercepted packets while the original packets are forwarded normally. The new flow is rerouted through the tunnel PIC back to the Packet Forwarding Engine for a route lookup, and then on to the content destination.

Configuration

To configure IPv6 FlowTapLite on an M120 router, perform these tasks:

CLI Quick Configuration

To quickly configure IPv6 FlowTapLite, copy the following commands and paste them into the CLI:

```
set system login class flowtap permissions flow-tap-operation
set system login user ftap uid 2000
set system login user ftap class flowtap
set system login user ftap authentication encrypted-password "xxxxxx"
set system services flow-tap-dtcp ssh
set interfaces vt-4/0/0 unit 0 family inet
set interfaces vt-4/0/0 unit 0 family inet6
set services flow-tap tunnel-interface vt-4/0/0.0
```

Configuring User Credentials

Step-by-Step Procedure

The username and password configured here are used by the mediation device when connecting and sending out DTCP requests.

1. Define a login class called **flowtap**:

```
[edit system]
user@router# set login class flowtap permissions flow-tap-operation
```

2. For the meditation device, configure a user called **ftap** with a unique identifier (UID):

```
[edit system]
user@router# set login user ftap uid 2000
```

3. Apply the **flowtap** class to the **ftap** user:

```
[edit system]
user@router# set login user ftap class flowtap
```

4. Configure the password used by the mediation device:

```
[edit system]
user@router# set login user ftap authentication encrypted-password xxxxxx
```

5. Commit the configuration:

```
[edit system]
user@router# commit
```


Configuring the Tunnel Interface for FlowTapLite

Step-by-Step Procedure

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.

1. Configure SSH from the **[edit system]** hierarchy level:

   ```
   [edit system]
   user@router# set services flow-tap-dtcp ssh
   ```
2. Commit the configuration:

   ```
   [edit system]
   user@router# commit
   ```

Configuring the Logical Tunnel Interface

Step-by-Step Procedure

1. Configure the logical interface and assign it to the dynamic flow control process (dfcd) at the **[edit interfaces]** hierarchy level:

   ```
   [edit interfaces]
   user@router# set vt-4/0/0 unit 0 family inet
   ```
2. Include the mandatory **inet6** statement:

   ```
   [edit interfaces]
   user@router# set vt-4/0/0 unit 0 family inet6
   ```
3. Commit the configuration:

   ```
   [edit interfaces]
   user@router# commit
   ```

Configuring FlowTapLite

Step-by-Step Procedure

1. Include the **flow-tap** statement and the tunnel interface at the **[edit services]** hierarchy level:

   ```
   [edit services]
   user@router# set flow-tap tunnel-interface vt-4/0/0.0
   ```
2. Commit the configuration:

   ```
   [edit services]
   user@router# commit
   ```

Results

Check the results of the configuration:

```
[edit]
user@router# show
system {
    [...Output Truncated...]
    login {
        class flowtap {
            permissions flow-tap-operation;
        }
        user ftap {
            uid 2000;
            class flowtap;
            authentication {
                encrypted-password "xxxxxx"; ## SECRET-DATA
```

```
            }
        }
    }
    services {
        telnet;
        flow-tap-dtcp {
            ssh;
        }
    }
}
interfaces {
    vt-4/0/0 {
        unit 0 {
            family inet;
            family inet6;
        }
    }
}
[...Output Truncated...]
services {
    flow-tap {
        tunnel-interface vt-4/0/0.0;
    }
}
```

Verification

To confirm that the configuration is working properly, perform the following tasks:

- Verifying That the Router Received the Filter Request on page 118
- Checking That Filters Are Installed and Working on the Router on page 118
- Sending a List Request on page 119

Verifying That the Router Received the Filter Request

Purpose After the mediation device sends the filters to the router, the mediation device must receive a message from the router confirming that the router has received the filter request.

Action Check that the mediation device has received a message similar to the one below:

```
DTCP/0.8 200 OK
SEQ: 1
CRITERIA-ID: 1
TIMESTAMP: 2009-09-29 06:12:05.725
AUTHENTICATION-INFO: 55f9dc3debd3c7356951410f165f2a9cc5606063
```

Meaning The message above is an example of a successfully received filter request.

Checking That Filters Are Installed and Working on the Router

Action Use the **show filter** and the **show filter index** commands to check that filters are installed:

```
ADPC2(diving vty)# show filter
Program Filters:
---------------
```



```
   Index      Dir     Cnt    Text     Bss  Name
--------   ------  ------  ------  ------  --------
       1      104       0      20      20  __default_bpdu_filter__
   17000       52       0       4       4  __default_arp_policer__
   57007      104     144      16      16  __flowtap_inet__
   65280       52       0       4       4  __auto_policer_template__
   65281      104       0      16      16  __auto_policer_template_1__
   65282      156       0      32      32  __auto_policer_template_2__
   65283      208       0      48      48  __auto_policer_template_3__
   65284      260       0      64      64  __auto_policer_template_4__
   65285      312       0      80      80  __auto_policer_template_5__
   65286      364       0      96      96  __auto_policer_template_6__
   65287      416       0     112     112  __auto_policer_template_7__
   65288      468       0     128     128  __auto_policer_template_8__
37748736   156   144   80   80 __ftaplite_filter__ifl__70__out__ipv6_
37748737   156   144   80   80 __ftaplite_filter__vrf__4__in__ipv6_
37748738   156   144   80   80 __ftaplite_filter__ifl__71__in__ipv6_
37748739   156   144   80   80 __ftaplite_filter__vrf__0__in__ipv6_

ADPC2(diving vty)# show filter index 37748738 counters
Filter Counters/Policers:
   Index              Packets                 Bytes  Name
--------   --------------------  --------------------  --------
37748738              8851815             601923420
__ftaplite_term_ftap_3___counter
```

Meaning The last four filters in the output for the **show filter** command above are the filters installed on the Packet Forwarding Engine. The **show filter index** command shows a non-zero packet count, indicating that the packets are hitting the filter.

Sending a List Request

Purpose To verify that the correct filters are installed in the Packet Forwarding Engine.

Action Use client software to send a list request to the Packet Forwarding Engine. In your list request, you can include the following three parameters individually or together: **CSource-Id**, **CDest-ID**, and **Criteria-ID**. With all requests, you must include the **CSource-Id**. Below is an example of a list request using the **CSource-Id**:

```
LIST DTCP/0.8
Csource-ID: ftap1
Flags: Both
```

Below is an example of a response:

```
DTCP/0.8 200 OK
SEQ: 51
TIMESTAMP: 2009-10-04 07:56:43.003
CRITERIA-ID: 1
CSOURCE-ID: ftap1
CDEST-ID: cd1
CSOURCE-ADDRESS: 10.209.152.15
FLAGS: Static
AVERAGE-BANDWIDTH: 0
MATCHING-PACKETS: 0
MATCHING-BYTES: 0
NUM-REFRESH: 0
LAST-REFRESH: 2009-10-04 07:54:30.870
```



```
X-JTAP-INPUT-INTERFACE: ge-2/1/1.0,ge-2/1/1.1,ge-2/1/1.2
SOURCE-ADDRESS: 203.0.113.1
DEST-ADDRESS: 192.168.0.1/32
SOURCE-PORT: 1000
DEST-PORT: 2000
PROTOCOL: 17
X-JTAP-CDEST-DEST-ADDRESS: 192.168.99.81
X-JTAP-CDEST-DEST-PORT: 8001
X-JTAP-CDEST-SOURCE-ADDRESS: 192.168.208.9
X-JTAP-CDEST-SOURCE-PORT: 34675
X-JTAP-CDEST-TTL: 64
CRITERIA-NUM: 1
CRITERIA-COUNT: 1
AUTHENTICATION-INFO: 0f49ff600a3d8d7d312c5031f74cc17540bc9200
```

You can also delete the request. Below is an example of a delete request:

```
DELETE DTCP/0.8
Csource-ID: ftap
CDest-ID: cd1
Flags: STATIC
```

Related Documentation

- Configuring FlowTapLite on page 112
- *flow-tap*
- *Tunnel Interface Configuration on MX Series Routers Overview*

Configuring the Interfaces That Will Be Actively Monitored

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 {
  so-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 7.

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:

Collecting Flow Records 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;
         }
      }
   }
}
```

Collecting Flow Records 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;
        }
      }
    }
  }
}
```

Collecting Flow Records with a Template

Flow monitoring version 9, which is based upon 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 “Version 9 Formats and Fields” on page 18.

```
[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 60 seconds and the
              # range is 10–600
            option-refresh-rate seconds 10; # The default is 60 seconds and the range
              # is 10–600 seconds.
            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 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;
        }
      }
    }
  }
}
```

Enabling the Monitoring Services, Adaptive Services, or Multiservices Interfaces and the Export Interface

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 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 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

- Flow Monitoring Overview on page 3
- Active Flow Monitoring Overview on page 91
- Active Flow Monitoring Applications on page 91
- Replicating Routing Engine-Based Sampling to Multiple Flow Servers on page 126

Routing Engine-Based Sampling 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: VRF Routing Engine-Based Sampling

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:

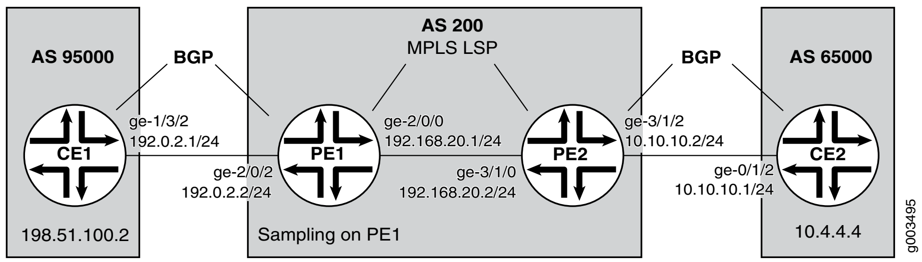
- Junos OS Release 10.1 or later
- M Series, MX Series, or T Series router

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

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).

Figure 21: Routing Engine-Based Sampling Network Topology



Configuration

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

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 on page 145
- Verifying Sampled Traffic on page 145
- Cross Verifying Sampled Traffic on page 146

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*" and ""Example: Flow Collector Interface Configuration" on page 78."

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 145 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 145 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 145 task, indicating that the intended traffic is being sampled.

Related Documentation

- *Configuring Traffic Sampling*
- *Flow Monitoring Feature Guide for Routing Devices*

Option: Configuring an Aggregate Export Timer

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;
    }
  }
}
```

Option: Configuring Port Mirroring

You can copy packets and reroute them to another interface by using port mirroring. To send packet copies to an interface, include the **interface** statement at the **[edit forwarding-options port-mirroring** family ***family-name*** **output]** hierarchy level and specify the interface to receive the traffic.

You can even send port-mirrored traffic to a monitoring services or adaptive services interface. If you choose this option, accepted traffic is copied and the packet copies are sent to the services interface for flow processing.

To configure how often packets are copied from the monitored traffic, include the **rate** statement at the **[edit forwarding-options port-mirroring** family ***family-name*** **input]** hierarchy level. A rate of **1** port-mirrors every packet, while a rate of **10** port-mirrors every tenth packet.

```
[edit]
forwarding-options {
  port-mirroring {
    family (inet | inet6) {
      input {
        rate 1;
      }
      output {
        interface sp-2/0/0.0;
      }
    }
  }
}
```

Option: Configuring Port Mirroring with Filter-Based Forwarding and a Monitoring Group

For active flow monitoring, you can load-balance traffic across multiple Monitoring Services PICs using the same method as passive flow monitoring. The only difference is that you do not configure the input interface with the **passive-monitor-mode** statement at the **[edit interfaces *interface-name*]** hierarchy level.

To load-balance traffic for active flow monitoring, port-mirror the incoming packets to a tunnel services interface. Redirect this copy of the traffic to a filter-based forwarding instance by applying a firewall filter to the tunnel services interface. Configure the instance to send the traffic to a group of monitoring services interfaces. Finally, use a monitoring group to send flow records from the monitoring services interfaces to a flow server.

NOTE: When you load-balance port-mirrored traffic across several Monitoring Services interfaces, there are some limitations:

- The original Monitoring Services PIC supports this method. You cannot use a Monitoring Services II PIC.
- You must use the suite of **show passive-monitoring** commands to monitor traffic. The **show services accounting** commands are not supported.
- Because load-balanced traffic is routed through the Tunnel Services PIC, the total throughput of the load-balanced traffic coming from the Monitoring Services PICs cannot exceed the bandwidth of the tunnel interface.

For detailed information on this method, see "Copying and Redirecting Traffic with Port Mirroring and Filter-Based Forwarding" on page 58.

Option: Sending Port-Mirrored Traffic to Multiple Export Interfaces by Using Next-Hop Groups

To send port-mirrored traffic to multiple flow servers or packet analyzers, you can use the **next-hop-group** statement. The router can make up to 16 copies of traffic per group and send the traffic to the next-hop group members you configure. A maximum of 30 groups can be configured on a router at any given time. The port-mirrored traffic can be sent to any interface, except aggregated SONET/SDH, aggregated Ethernet, loopback (**lo0**), or administrative (**fxp0**) interfaces. To configure multiple port mirroring with next-hop groups, include the **next-hop-group** statement at the **[edit forwarding-options]** hierarchy level.

You must port-mirror the initial traffic to a tunnel interface so that it can be filtered and duplicated. Also, you need configure only the interface names for point-to-point interfaces, but you must configure the interface names and a next hop for multipoint interfaces (such as Ethernet).

```
[edit]
forwarding-options {
  port-mirroring {
    family inet {
      input {
        rate 1;
      }
      output {
        interface vt-3/3/0.1;
        no-filter-check;
      }
    }
  }
  next-hop-group ftp-traffic {
```



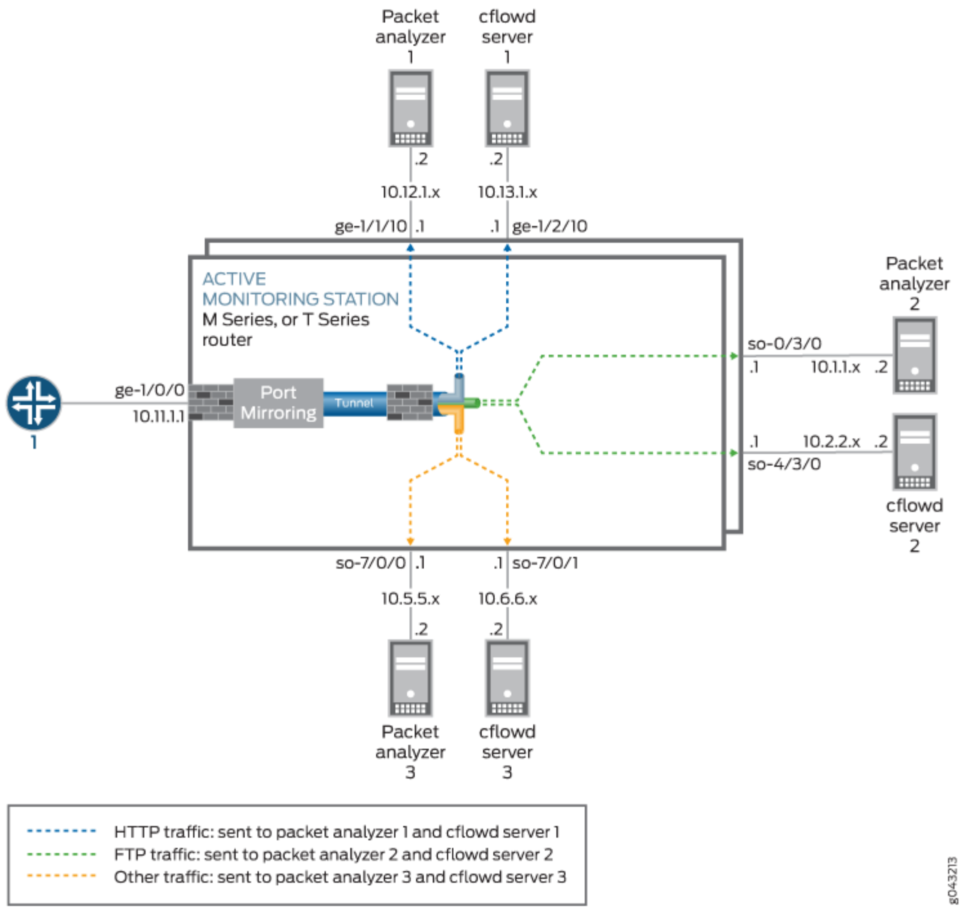
```
    interface so-4/3/0.0;
    interface so-0/3/0.0;
  }
  next-hop-group http-traffic {
    interface ge-1/1/0.0 {
      next-hop 10.12.1.2;
    }
    interface ge-1/2/0.0 {
      next-hop 10.13.1.2;
    }
  }
  next-hop-group default-collect {
    interface so-7/0/0.0;
    interface so-7/0/1.0;
  }
}
```

NOTE: **Next-hop groups** are supported on M Series routers only, except the M120 router and the M320 router.

Example: Multiple Port Mirroring with Next-Hop Groups Configuration

When you need to analyze traffic containing more than one packet type, or you wish to perform multiple types of analysis on a single type of traffic, you can implement multiple port mirroring and next-hop groups. You can make up to 16 copies of traffic per group and send the traffic to next-hop group members. A maximum of 30 groups can be configured on a router at any given time. The port-mirrored traffic can be sent to any interface, except aggregated SONET/SDH, aggregated Ethernet, loopback (**lo0**), or administrative (**fxp0**) interfaces. To send port-mirrored traffic to multiple flow servers or packet analyzers, you can use the **next-hop-group** statement at the **[edit forwarding-options]** hierarchy level.

Figure 22: Active Flow Monitoring—Multiple Port Mirroring with Next-Hop Groups Topology Diagram



"Active Flow Monitoring" on page 89 shows an example of how to configure multiple port mirroring with next-hop groups. All traffic enters the monitoring router at interface **ge-1/0/0**. A firewall filter counts and port-mirrors all incoming packets to a Tunnel Services PIC. A second filter is applied to the tunnel interface and splits the traffic into three categories: HTTP traffic, FTP traffic, and all other traffic. The three types of traffic are assigned to three separate next-hop groups. Each next-hop group contains a unique pair of exit interfaces that lead to different groups of packet analyzers and flow servers.

NOTE: Instances enabled to mirror packets to different destinations from the same PFE, also use different sampling parameters for each instance. When we configure Layer2 Port-mirroring with both global port-mirroring and instance based port-mirroring, PIC level instances will override FPC level and the FPC level will override the Global instance.

```
[edit]
interfaces {
```



```
ge-1/0/0 { # This is the input interface where packets enter the router.
  unit 0 {
    family inet {
      filter {
        input mirror_pkts; # Here is where you apply the first filter.
      }
      address 10.11.1.1/24;
    }
  }
}
ge-1/1/0 { # This is an exit interface for HTTP packets.
  unit 0 {
    family inet {
      address 10.12.1.1/24;
    }
  }
}
ge-1/2/0 { # This is an exit interface for HTTP packets.
  unit 0 {
    family inet {
      address 10.13.1.1/24;
    }
  }
}
so-0/3/0 { # This is an exit interface for FTP packets.
  unit 0 {
    family inet {
      address 10.1.1.1/30;
    }
  }
}
so-4/3/0 { # This is an exit interface for FTP packets.
  unit 0 {
    family inet {
      address 10.2.2.1/30;
    }
  }
}
so-7/0/0 { # This is an exit interface for all remaining packets.
  unit 0 {
    family inet {
      address 10.5.5.1/30;
    }
  }
}
so-7/0/1 { # This is an exit interface for all remaining packets.
  unit 0 {
    family inet {
      address 10.6.6.1/30;
    }
  }
}
vt-3/3/0 { # The tunnel interface is where you send the port-mirrored traffic.
  unit 0 {
    family inet;
  }
```



```
        unit 1 {
            family inet {
                filter {
                    input collect_pkts;  # This is where you apply the second firewall filter.
                }
            }
        }
    }
}
forwarding-options {
    port-mirroring { # This is required when you configure next-hop groups.
        family inet {
            input {
                rate 1; # This port-mirrors all packets (one copy for every packet received).
            }
            output { # Sends traffic to a tunnel interface to enable multiport mirroring.
                interface vt-3/3/0.1;
                no-filter-check;
            }
        }
    }
    next-hop-group ftp-traffic { # Point-to-point interfaces require you to specify the
        interface so-4/3/0.0; # interface name.
        interface so-0/3/0.0;
    }
    next-hop-group http-traffic { # Configure a next hop for all multipoint interfaces.
        interface ge-1/1/0.0 {
            next-hop 10.12.1.2;
        }
        interface ge-1/2/0.0 {
            next-hop 10.13.1.2;
        }
    }
    next-hop-group default-collect {
        interface so-7/0/0.0;
        interface so-7/0/1.0;
    }
}
firewall {
    family inet {
        filter mirror_pkts { # Apply this filter to the input interface.
            term catch_all {
                then {
                    count input_mirror_pkts;
                    port-mirror; # This action sends traffic to be copied and port-mirrored.
                }
            }
        }
        filter collect_pkts { # Apply this filter to the tunnel interface.
            term ftp-term { # This term sends FTP traffic to an FTP next-hop group.
                from {
                    protocol ftp;
                }
                then next-hop-group ftp-traffic;
            }
            term http-term { # This term sends HTTP traffic to an HTTP next-hop group.
```

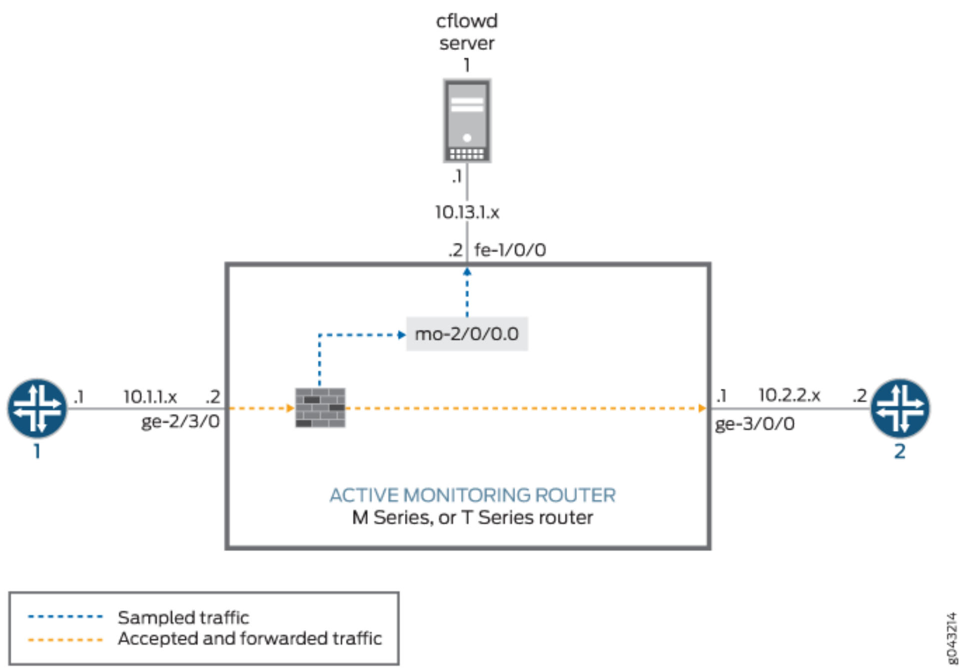
```
            from {
              protocol http;
            }
            then next-hop-group http-traffic;
          }
          term default { # This sends all remaining traffic to a final next-hop group.
            then next-hop-group default-collectors;
          }
        }
      }
    }
```

Related Documentation

- *Understanding Port Mirroring*
- *Configuring Port Mirroring*

Example: Sampling Configuration

Figure 23: Active Flow Monitoring—Sampling Configuration Topology Diagram



In Figure 23 on page 154, 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 SONET/SDH, Gigabit Ethernet, and so on). The export interface leading to the flow server is **fe-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;
        }
      }
    }
  }
  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 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%
```

Example: Sampling Instance Configuration

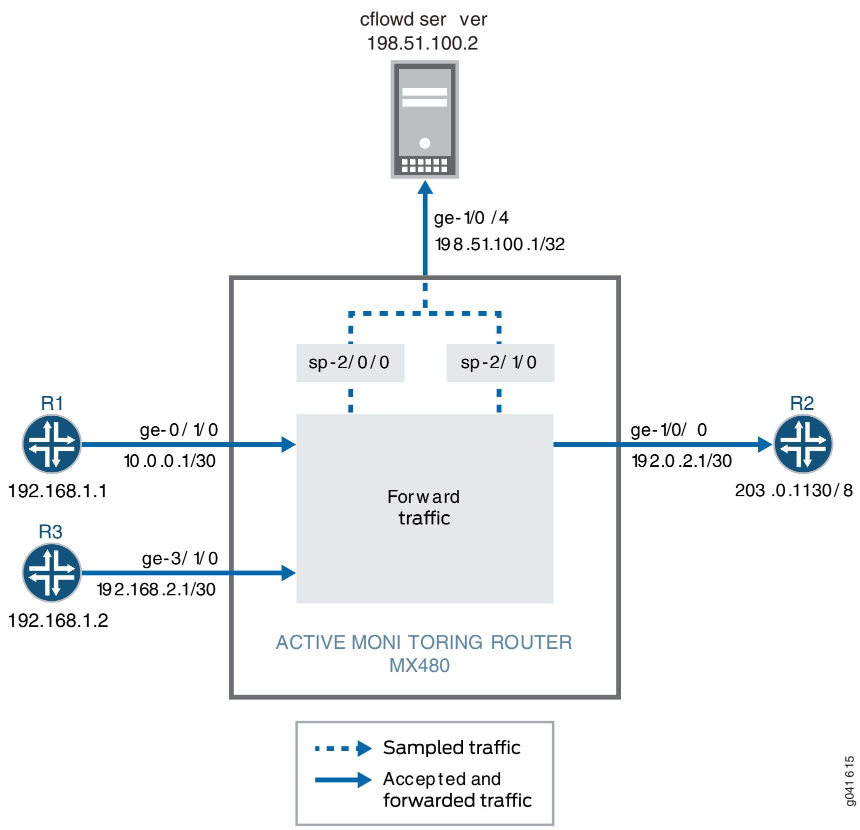
You can configure active sampling using a sampling instance and associate that sampling instance to a particular FPC, MPC, or DPC. In addition, you can define multiple sampling instances associated with multiple destinations and protocol families per sampling instance destination.

- Example Network Details on page 158
- Example Router Configuration on page 160
- Configuration Commands Used for the Configuration Example on page 162
- Verifying Your Work on page 163

Example Network Details

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

Figure 24: Active Flow Monitoring—Sampling Instance Configuration Topology Diagram



In Figure 24 on page 159, 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
```

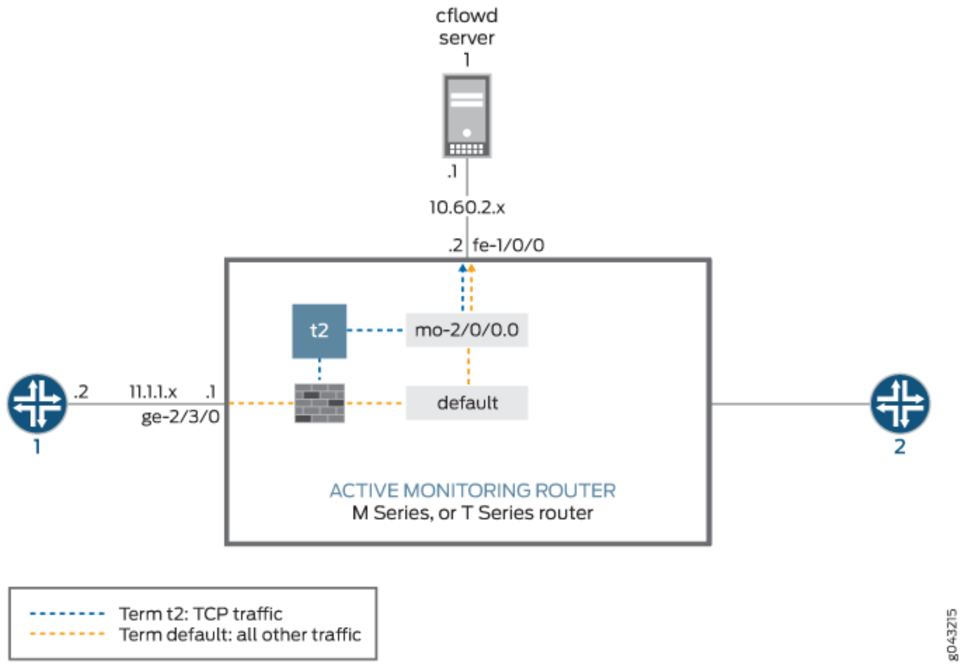
Related Documentation

- *Sampling Instance Configuration*
- Configuring Active Flow Monitoring on page 102
- *sampling-instance*

Example: Sampling and Discard Accounting Configuration

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 25: Active Flow Monitoring—Sampling and Discard Accounting Topology Diagram



In Figure 25 on page 165, 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 10.1.1.1/20;
            }
        }
    }
}
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
```


PART 4

Operational Commands

CHAPTER 9

Active Flow Monitoring Commands

show forwarding-options next-hop-group

Syntax

show forwarding-options next-hop-group
<terse | brief | detail>
<*group-name*>

Release Information

Command introduced in Junos OS Release 9.6.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Support for IPv6 introduced in Junos OS Release 14.2 for the MX Series routers.

Description

Display current state of next-hop groups.

Options

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

group-name—(Optional) Display a single next-hop group.

Required Privilege Level

view

Related Documentation

- show forwarding-options port-mirroring on page 177

List of Sample Output

show forwarding-options next-hop-group terse on page 175
show forwarding-options next-hop-group brief on page 175
show forwarding-options next-hop-group detail on page 175

Output Fields

Table 34 on page 174 lists the output fields for the **show forwarding-options next-hop-group** command. Output fields are listed in the approximate order in which they appear.

Table 34: show forwarding-options next-hop-group Output Fields

Field Name	Field Description	Level of Output
Next-hop-group	Name of next-hop group.	All levels
Type	Next-hop group type, such as **inet**, **inet6** or **layer-2**.	All levels
State	Next-hop group state, either **up** or **down**.	All levels
Members Interfaces	Names of interfaces to which next-hop group members belong.	**brief detail**
Member Subgroup	Names of subgroups to which next-hop group members belong.	**brief detail**
Number of members configured	Number of next-hop group members configured.	**detail**
Number of members that are up	Number of next-hop group members that are up.	**detail**

Table 34: show forwarding-options next-hop-group Output Fields *(continued)*

Field Name	Field Description	Level of Output
Number of subgroups configured	Number of subgroups configured.	**detail**
Number of subgroups that are up	Number of subgroups that are up.	**detail**

Sample Output

show forwarding-options next-hop-group terse

```
user@host> show forwarding-options next-hop-group terse
Next-hop-group                   Type             State
  nhg                              inet             up
  nhg6                             inet6            up
  vpls_nhg_2                     layer-2          down
```

show forwarding-options next-hop-group brief

```
user@host> show forwarding-options next-hop-group brief

Next-hop-group: nhg
      Type:  inet
      State: up
      Members Interfaces:
        ge-0/2/8.0             next-hop  192.0.2.10
        ge-5/1/8.0             next-hop  198.51.100.10
        ge-5/1/9.0             next-hop  203.0.113.10

Next-hop-group: nhg6
      Type:  inet6
      State: up
      Members Interfaces:
        ge-5/1/5.0             next-hop  2001:db8::1:10
        ge-5/1/6.0             next-hop  2001:db8::20:10        Member Subgroup:
 nhsg6
          Members Interfaces:
            ge-5/0/4.0             next-hop  2001:db8::3:1
            ge-5/1/4.0             next-hop  2001:db8::4:1

Next-hop-group: vpls_nhg_2
Type: layer-2        State: down
```

show forwarding-options next-hop-group detail

```
user@host> show forwarding-options next-hop-group detail

Next-hop-group: nhg
Type: inet
State: up
Number of members configured    : 3
Number of members that are up   : 3
Number of subgroups configured  : 0
Number of subgroups that are up : 0
```

```
Members Interfaces:                                 State
  ge-0/2/8.0             next-hop  192.0.2.10        up
  ge-5/1/8.0             next-hop  203.0.113.10        up
  ge-5/1/9.0             next-hop  198.51.100.10.10        up

Next-hop-group: nhg6
Type: inet6
State: up
Number of members configured    : 2
Number of members that are up   : 2
Number of subgroups configured  : 1
Number of subgroups that are up : 1
Members Interfaces:                                 State
ge-5/1/5.0             next-hop  2001:db8::1:10       up
ge-5/1/6.0             next-hop  2001:db8::20:10      up
Member Subgroup: nhsg6                            up
       Number of members configured    : 2
       Number of members that are up   : 2
       Members Interfaces:                                State
           ge-5/0/4.0             next-hop  2001:db8::3:1   up
           ge-5/1/4.0             next-hop  2001:db8::4:1   up

Next-hop-group: vpls_nhg_2
Number of members configured    : 2
Number of members that are up   : 0
Number of subgroups configured  : 0
Number of subgroups that are up : 0
Type: layer-2       State: down
Members Interfaces:       State
   ge-2/2/1.100        down
   ge-2/3/9.0       down
```

show forwarding-options port-mirroring

Syntax

show forwarding-options port-mirroring
<terse | detail>
<*instance-name*>

Release Information

Command introduced in Junos OS Release 9.6.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Display current state of port-mirroring instances.

Options

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

instance-name—(Optional) Display a single port-mirroring instance.

Required Privilege Level

view

Related Documentation

List of Sample Output

show forwarding-options port-mirroring terse on page 178
show forwarding-options port-mirroring detail on page 178

Output Fields

Table 35 on page 177 lists the output fields for the **show forwarding-options port-mirroring** command. Output fields are listed in the approximate order in which they appear.

Table 35: show forwarding-options port-mirroring Output Fields

Field Name	Field Description	Level of Output
Instance Name	Name of port-mirroring instance.	All levels
Instance Id	Instance identification number.	All levels
State	Instance state, either **up** or **down**.	All levels
Input parameters		
Rate	Rate (ratio of packets sampled).	**detail**
Run-length	Run length (number of consecutive packets sampled).	**detail**
Maximum-packet-length	Maximum packet length.	**detail**
Output parameters		
Family	Protocol family.	**detail**
State	Instance state, either **up** or **down**.	**detail**
Destination	Destination (next-hop group name).	**detail**

Sample Output

show forwarding-options port-mirroring terse

```
user@host> show forwarding-options port-mirroring terse
Instance Name        Instance Id    State
&global_instance       1      up
inst1         2      up
```

show forwarding-options port-mirroring detail

```
user@host> show forwarding-options port-mirroring detail
Instance Name: &global_instance
Instance Id: 1      State: up
  Input parameters:
     Rate:      10
     Run-length:     4
     Maximum-packet-length: 0
  Output parameters:
     Family: inet   State: up Destination: inet_nhg
     Family: vpls/eth-switch  State: up     Destination: vpls_nhg

Instance Name: inst1
Instance Id: 2      State: up
  Input parameters:
     Rate:      1
     Run-length:     0
     Maximum-packet-length: 200
  Output parameters:
     Family: inet         State: up     Destination: inet_nhg
     Family: vpls/eth-switch  State: down   Destination: vpls_nhg_2
```


show services accounting aggregation

Syntax

show services accounting aggregation *aggregation-type <aggregation-value>*
<detail | extensive | terse>
<limit *limit-value*>
< name *service-name*>
<order (bytes | packets)>

Release Information

Command introduced before Junos OS Release 7.4.

Description

Display information about the aggregated active flows being processed by the accounting service.

Options

aggregation-type <aggregation-value>—Display information for the specified aggregation type and optional value:

- **as** ***<source-as-value | destination-as-value | input-snmp-interface-index-value | output-snmp-interface-index-value>***—Aggregate by autonomous system (AS).
- **destination-prefix** ***<destination-prefix-value | destination-as-value | output-snmp-interface-index-value>***—Aggregate by destination prefix.
- **protocol-port** ***<protocol-value | source-port-value | destination-port-value>***—Aggregate by protocol and port.
- **source-destination-prefix** ***<source-prefix-value | destination-prefix-value | destination-as-value | source-as-value | input-snmp-interface-index-value | output-snmp-interface-index-value>***—Aggregate by source and destination prefix.
- **source-prefix** ***<source-prefix-value | source-as-value | input-snmp-interface-index-value>***—Aggregate by source prefix.

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

limit ***limit-value***—(Optional) Limit the display output to the specified number of flows. The default is no limit.

name ***service-name***—(Optional) Display information about the aggregated flows for a specified service name.

order (bytes | packets)—(Optional) Display the flow with the ordering of the highest number, either by byte count or by packet count.

Additional Information

For information about aggregation configuration options, see the *Junos OS Services Interfaces Library for Routing Devices*.

Required Privilege Level

view

List of Sample Output

show services accounting aggregation protocol-port detail on page 181
show services accounting aggregation source-destination-prefix on page 181

show services accounting aggregation source-destination- prefix order packet detail on page 181
show services accounting aggregation source-destination- prefix extensive limit on page 182
show services accounting aggregation source-destination-prefix name terse on page 182

Output Fields

Table 36 on page 180 lists the output fields for the **show services accounting aggregation** command. Output fields are listed in the approximate order in which they appear.

Table 36: show services accounting aggregation Output Fields

Field Name	Field Description
Service Accounting interface	Name of the service accounting interface.
Local interface index	Index corresponding to the service accounting interface.
Service name	Name of a service that was configured at the [**edit forwarding-options accounting]** hierarchy level. The default display, **(default sampling)**, indicates the service was configured at the [**edit forwarding-options sampling-level**] hierarchy level.
Protocol	Protocol identifier and number.
Source Port	Source port identifier and number.
Destination Port	Destination port identifier and number.
Source-AS	Source autonomous system (AS) number.
Destination-AS	Destination AS number.
Source Prefix	Source prefix.
Destination Prefix	Destination prefix.
Source address	Source address.
Source prefix length	Source prefix length.
Destination address	Destination address.
Destination prefix length	Destination prefix length.
Input SNMP interface index	SNMP index of the interface the packet came in on.
Output SNMP interface index	SNMP index of the interface the packet went out on.

Table 36: show services accounting aggregation Output Fields *(continued)*

Field Name	Field Description
Start time	Actual time when the packet in this aggregation was first seen.
End time	Actual time when the packet in this aggregation was last seen.
Flow count	Number of flows in the aggregation.
Packet count	Number of packets in the aggregation.
Byte count	Number of bytes in the aggregation.

Sample Output

show services accounting aggregation protocol-port detail

```
user@host> show service accounting aggregation protocol-port detail
Service Accounting interface: mo-2/0/0, Local interface index: 468
Service name: (default sampling)
  Protocol: 6, Source port: 20, Destination port: 20
  Start time: 442349, End time: 6425714
  Flow count: 194, Packet count: 4294964388, Byte count: 4294781184

  Protocol: 0, Source port: 0, Destination port: 0
  Start time: 442349, End time: 6425749
  Flow count: 204, Packet count: 4294964324, Byte count: 4294777088

  Protocol: 17, Source port: 123, Destination port: 123
  Start time: 442364, End time: 6425784
  Flow count: 186, Packet count: 4294964152, Byte count: 4294766080
```

show services accounting aggregation source-destination-prefix

```
user@host> show service accounting aggregation source-destination-prefix
Service Accounting interface: rsp0, Local interface index: 171
Service name: (default sampling)
Interface state: Accounting
Source         Destination      Input        Output       Flow   Packet  Byte
prefix         prefix           interface    interface    count  count   count
192.0.2.0/20  198.51.100.0/24  ge-5/0/1.0  ge-5/0/0.0  256    491761  31472704
192.0.2.0/20   203.0.113.36/32  ge-5/0/1.0   ge-5/0/0.0   1      1926     123264
192.0.2.0/20   203.0.113.59/32  ge-5/0/1.0   ge-5/0/0.0   1      1926     123264
192.0.2.0/20   192.168.0.63/32  ge-5/0/1.0   ge-5/0/0.0   1      1925     123200
192.0.2.0/20   192.168.0.32/32  ge-5/0/1.0   ge-5/0/0.0   1      1925
```

show services accounting aggregation source-destination- prefix order packet detail

```
user@host> show service accounting aggregation source-destination-prefix order packet detail
name t2 input-snmp-interface-index 538
Service Accounting interface: mo-2/0/0, Local interface index: 468
Service name: t2
Source         Destination  Input SNMP  Output  SNMP   Flow   Packet Byte
Prefix            Prefix        Index      Index Count  Count    Count
10.1.1.2/20   192.168.167.1/0  538          432    1     60     46483
10.1.1.2/20   192.168.168.1/0  538          432    1     60      5191
10.1.1.2/20   192.168.154.1/0  538          432    2     60     45504
```

```
10.1.1.2/20   192.168.76.1/0   538         432    1      60     42177
10.1.1.2/20   192.168.149.1/0  538         432    1      60     49184
10.1.1.2/20   192.168.113.1/0  538         432    2      60     48757
```

show services accounting aggregation source-destination- prefix extensive limit

user@host> **show service accounting aggregation source-destination-prefix name t2 extensive limit 3**

```
Service Accounting interface: mo-2/0/0, Local interface index: 542
Service name: t2

  Source address: 10.1.1.2, Source prefix length: 20
  Destination address: 192.168.200.176.1, 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: 192.168.160.1, 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: 192.168.160.1, 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
```

show services accounting aggregation source-destination-prefix name terse

user@host> **show service accounting aggregation source-destination-prefix name T3 terse**

```
Service Accounting interface: rsp0, Local interface index: 171
Service name: T3
Interface state: Accounting
Source        Destination       Input        Output       Flow   Packet  Byte
prefix        prefix            interface    interface    count  count   count
10.1.0.0/20   192.168.3.0/24   ge-5/0/1.0   ge-5/0/0.0   256    639822  40948608
10.1.0.0/20   192.168.2.67/32   ge-5/0/1.0   ge-5/0/0.0   1      2485    159040
10.1.0.0/20   192.168.2.92/32   ge-5/0/1.0   ge-5/0/0.0   1      2485
```

show services accounting aggregation template

Syntax

show services accounting aggregation template
<template-name *template-name*>

Release Information

Command introduced in Junos OS Release 8.3.

Description

Display information for flow aggregation version 9 templates.

Options

none—Display information for all flow aggregation version version 9 templates.

template-name *template-name*—(Optional) Display information for the specified template only.

Required Privilege Level

view

List of Sample Output

show services accounting aggregation template template-name on page 183

Output Fields

Table 37 on page 183 lists the output fields for the **show services accounting aggregation template** command. Output fields are listed in the approximate order in which they appear.

Table 37: show services accounting aggregation template Output Fields

Field Name	Field Description
MPLS Label 1	Position of first MPLS label.
MPLS Label 2	Position of second MPLS label.
MPLS Label 3	Position of third MPLS label.
MPLS Top Level Address	Outer top label FEC IP address.
Packet Count	Number of packets sent.

Sample Output

show services accounting aggregation template template-name

```
user@host> show services accounting aggregation template template-name mpls
MPLS label 1: 299808, MPLS label 2: 0, MPLS label 3: 0
Source address: 192.0.2.2, Destination address: 10.255.15.22, Top Label Address:
 198.51.100.10
Source port: 0, Destination port: 0
Protocol: 61, TOS: 0, TCP flags: 0
Source mask: 24, Destination mask: 32
Input SNMP interface index: 503, Output SNMP interface index: 505
Start time: 40780, End time: 157330
Packet count: 3949198, Byte count: 181663062
```

show services accounting errors

Syntax

show services accounting errors
<inline-jflow | name (* | all | *service-name*)>

Release Information

Command introduced before Junos OS Release 7.4.

Description

Display active flow error statistics.

Options

none—Display error statistics for all services accounting instances.

inline-jflow fpc-slot *slot-number*—(Optional) Display error statistics for inline jflow.

name (* | all | *service-name*)—(Optional) Display active flow error statistics. Use a wildcard character, specify all services, or provide a specific service name.

Required Privilege Level

view

Related Documentation

- show services accounting flow on page 189

List of Sample Output

show services accounting errors (Monitoring PIC interface) on page 186
show services accounting errors (Service PIC interface) on page 187
show services accounting errors inline-jflow fpc-slot (When Only IPv6 Is Configured) on page 187
show services accounting errors inline-jflow fpc-slot (When IPv4, IPv6, and VPLS Are Configured) on page 187
show services accounting errors inline-jflow (MX80 Router When Both IPv4 and IPv6 Are Configured) on page 187
show services accounting errors inline-jflow fpc-slot(PTX1000 Router When Both IPv4 and IPv6 Are Configured) on page 188

Output Fields

Table 38 on page 185 lists the output fields for the **show services accounting errors** command. Output fields are listed in the approximate order in which they appear.

Table 38: show services accounting errors Output Fields

Field	Field Description
Service Accounting interface	Name of the service accounting interface.
Local interface index	Index counter of the local interface.
FPC slot	Slot number of the FPC for which the flow information is displayed. (Available only when the **inline-jflow fpc-slot *slot-number*** option is used.)
Service name	Name of a service that was configured at the [**edit forwarding-options accounting**] hierarchy level. The default display, **(default sampling)**, indicates the service was configured at the [**edit forwarding-options sampling-level**] hierarchy level.

Table 38: show services accounting errors Output Fields *(continued)*

Field	Field Description
Error Information	
Packets dropped (no memory)	Number of packets dropped because of memory shortage.
Packets dropped (not IP)	Number of non-IP packets dropped.
Packets dropped (not IPv4)	Number of packets dropped because they failed the IPv4 version 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 free failures.
Memory free list failures	Number of flow records received from the free list that failed. Memory is nearly exhausted, or too many new flows greater than 128 KB are being created per second.
Memory overload	Whether the memory has been overloaded. The response can be **Yes** or **No**.
PPS overload	Whether the PIC is receiving more packets per second than the configured threshold. The response can be **Yes** or **No**.
BPS overload	Whether the PIC is receiving more bits per second than the configured threshold. The response can be **Yes** or **No**.
Flow Creation Failures	Number of times flow creation failed.
Route Record Lookup Failures	Number of times the route record lookup failed.
AS Lookup Failures	Number of times autonomous system lookup failed.
Export Packet Failures	Number of times packet export failed.

Sample Output

show services accounting errors (Monitoring PIC interface)

```
user@host> show services accounting errors
Service Accounting interface: mo-1/1/0, Local interface index: 15
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: No
```

Sample Output

show services accounting errors (Service PIC interface)

```
user@host> show services accounting errors
Service Accounting interface: sp-0/1/0
Service name: (default sampling)
  Error information
    Service sets dropped: 0, Active timeout failures: 0
    Export packet failures: 0, Flow creation failures: 0
    Memory overload: No

Service Accounting interface: sp-1/0/0
Service name: (default sampling)
  Error information
    Service sets dropped: 0, Active timeout failures: 0
    Export packet failures: 0, Flow creation failures: 0
    Memory overload: No
```

show services accounting errors inline-jflow fpc-slot (When Only IPv6 Is Configured)

```
user@host> show services accounting errors inline-jflow fpc-slot 5
  Error information
    FPC Slot: 5
    Flow Creation Failures: 0
    Route Record Lookup Failures: 0, AS Lookup Failures: 0
    Export Packet Failures: 0
    Memory Overload: No, Memory Alloc Fail Count: 0
```

show services accounting errors inline-jflow fpc-slot (When IPv4, IPv6, and VPLS Are Configured)

```
user@host> show services accounting errors inline-jflow fpc-slot 5
  Error information
    FPC Slot: 5
    Flow Creation Failures: 0
    Route Record Lookup Failures: 0, AS Lookup Failures: 0
    Export Packet Failures: 0
    Memory Overload: No, Memory Alloc Fail Count: 0

    IPv4:
    IPv4 Flow Creation Failures: 0
    IPv4 Route Record Lookup Failures: 0, IPv4 AS Lookup Failures: 0
    IPv4 Export Packet Failures: 0

    IPv6:
    IPv6 Flow Creation Failures: 0
    IPv6 Route Record Lookup Failures: 0, IPv6 AS Lookup Failures: 0
    IPv6 Export Packet Failures: 0


    VPLS:
    VPLS Flow Creation Failures: 0
    VPLS Export Packet Failures: 0
```

show services accounting errors inline-jflow (MX80 Router When Both IPv4 and IPv6 Are Configured)

```
user@host> show services accounting errors inline-jflow
  Error information
    TFEB Slot: 0
```



```
 Flow Creation Failures: 0
 Route Record Lookup Failures: 0, AS Lookup Failures: 0
 Export Packet Failures: 0
 Memory Overload: No

IPv4:
 IPv4 Flow Creation Failures: 0
 IPv4 Route Record Lookup Failures: 0, IPv4 AS Lookup Failures: 0
 IPv4 Export Packet Failures: 0

 IPv6:
 IPv6 Flow Creation Failures: 0
 IPv6 Route Record Lookup Failures: 0, IPv6 AS Lookup Failures: 0
 IPv6 Export Packet Failures: 0
```

show services accounting errors inline-jflow fpc-slot (PTX1000 Router When Both IPv4 and IPv6 Are Configured)

```
user@host> show services accounting errors inline-jflow fpc-slot 0
Error information
FPC Slot: 0
Flow Creation Failures: 0
Route Record Lookup Failures: 0, AS Lookup Failures: 0
Export Packet Failures: 0
Memory Overload: No, Memory Alloc Fail Count: 0

IPv4:
IPv4 Flow Creation Failures: 0
IPv4 Route Record Lookup Failures: 0, IPv4 AS Lookup Failures: 0
IPv4 Export Packet Failures: 0

IPv6:
IPv6 Flow Creation Failures: 0
IPv6 Route Record Lookup Failures: 0, IPv6 AS Lookup Failures: 0
IPv6 Export Packet Failures: 0
```

show services accounting flow

Syntax

show services accounting flow
<inline-jflow fpc-slot *slot-number* | logical-system (all | *logical-system*) | name (* | all | *service-name*)>

Release Information

Command introduced before Junos OS Release 7.4.
Junos OS Release 10.0 added the capability to display output from multiple sampling instances.

Description

Display active flow statistics.

Options

none—Display active flow statistics for all service instances.

logical-system (all | *logical-system*)—(Optional) Display active flow statistics for the specified logical system or all logical systems on the device.

inline-jflow (fpc-slot *slot-number*)—(Optional) Display inline flow statistics for the specified FPC.

name (* | all | *service-name*)—(Optional) Display services accounting active flow statistics. Use a wildcard character, specify all services, or provide a specific service name.

Required Privilege Level

view

Related Documentation

- show services accounting status on page 203

List of Sample Output

show services accounting flow (Flow Aggregation v5/v8 Configuration) on page 190
show services accounting flow (Flow Aggregation v9 Configuration) on page 191
show services accounting flow name on page 191
show services accounting flow name all on page 191
show services accounting flow (Multiple Sampling Instances) on page 192
show services accounting flow inline-jflow fpc-slot (for IPv4 Flow) on page 192
show services accounting flow inline-jflow fpc-slot (with IPv4, IPv6, and VPLS Configuration) on page 192
show services accounting flow inline-jflow (MX80 Router with IPv4 and IPv6 Configuration) on page 193
show services accounting flow inline-jflow fpc-slot (PTX1000 Router When Both IPv4 and IPv6 Are Configured) on page 193

Output Fields

Table 39 on page 189 lists the output fields for the **show services accounting flow** command. Output fields are listed in the approximate order in which they appear.

Table 39: show services accounting flow Output Fields

Output Field	Output Field Description
Service Accounting interface	Name of the service accounting interface.

Table 39: show services accounting flow Output Fields *(continued)*

Output Field	Output Field Description
Local interface index	Index counter of the local interface.
Service name	Name of a service that was configured at the **[edit forwarding-options accounting]** hierarchy level. The default display, **(default sampling)**, indicates the service was configured at the [**edit forwarding-options sampling-level**] hierarchy level.
Flow Information	
FPC Slot	Slot number of the FPC for which the flow information is displayed. (Available only when the **inline-jflow fpc-slot** ***slot-number*** option is used.)
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 cflowd packets exported by an operational PIC.
Flows inactive timed out	Total number of flows that are exported because of inactivity.
Flows active timed out	Total number of long-lived flows that are exported because of an active timeout.

Sample Output

show services accounting flow (Flow Aggregation v5/v8 Configuration)

```
user@host> show services accounting flow
Service Accounting interface: rsp0, Local interface index: 171
Service name: (default sampling)
Interface state: Accounting
  Flow information
    Flow packets: 87168293, Flow bytes: 5578770752
    Flow packets 10-second rate: 45762, Flow bytes 10-second rate: 2928962
    Active flows: 1000, Total flows: 2000
    Flows exported: 19960, Flows packets exported: 582
    Flows inactive timed out: 1000, Flows active timed out: 29000
```

show services accounting flow (Flow Aggregation v9 Configuration)

```
user@host> show services accounting flow
  Flow information
    Service Accounting interface: sp-7/1/0, Local interface index: 149
    Flow packets: 0, Flow bytes: 0
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 0
    Flows exported: 0, Flows packets exported: 1
    Flows inactive timed out: 0, Flows active timed out: 0
```

show services accounting flow name

```
user@host> show services accounting flow name count2
Service Accounting interface: mo-1/1/0, Local interface index: 15
Service name: count2
  Flow information
    Flow packets: 0, Flow bytes: 0
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 0
    Flows exported: 0, Flows packets exported: 0
    Flows inactive timed out: 0, Flows active timed out: 0
```

show services accounting flow name all

```
user@host> show services accounting flow name all
Service Accounting interface: rsp0, Local interface index: 171
Service name: T2
Interface state: Accounting
  Flow information
    Flow packets: 37609891, Flow bytes: 2407033024
    Flow packets 10-second rate: 45762, Flow bytes 10-second rate: 2928953
    Active flows: 1000, Total flows: 1000
    Flows exported: 6705, Flows packets exported: 198
    Flows inactive timed out: 0, Flows active timed out: 13000

Service Accounting interface: rsp0, Local interface index: 171
Service name: T3
Interface state: Accounting
  Flow information
    Flow packets: 37750807, Flow bytes: 2416051712
    Flow packets 10-second rate: 45762, Flow bytes 10-second rate: 2928940
    Active flows: 1000, Total flows: 1000
    Flows exported: 13437, Flows packets exported: 378
    Flows inactive timed out: 0, Flows active timed out: 13000

Service Accounting interface: rsp0, Local interface index: 171
Service name: T4
 Interface state: Accounting
  Flow information
    Flow packets: 0, Flow bytes: 0
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 0
    Flows exported: 0, Flows packets exported: 0
    Flows inactive timed out: 0, Flows active timed out: 0

Service Accounting interface: rsp0, Local interface index: 171
Service name: count1
Interface state: Accounting
  Flow information
    Flow packets: 0, Flow bytes: 0
```

```
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 0
    Flows exported: 0, Flows packets exported: 0
    Flows inactive timed out: 0, Flows active timed out: 0
```

show services accounting flow (Multiple Sampling Instances)

```
user@host> show services accounting flow
  Flow information
    Service Accounting interface: sp-2/0/0, Local interface index: 215
    Flow packets: 9867, Flow bytes: 631488
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 628
    Active flows: 2, Total flows: 10
    Flows exported: 4028, Flows packets exported: 6150
    Flows inactive timed out: 8, Flows active timed out: 4026

    Service Accounting interface: sp-2/1/0, Local interface index: 223
    Flow packets: 0, Flow bytes: 0
    Flow packets 10-second rate: 0, Flow bytes 10-second rate: 0
    Active flows: 0, Total flows: 0
    Flows exported: 0, Flows packets exported: 1
    Flows inactive timed out: 0, Flows active timed out: 0
```

show services accounting flow inline-jflow fpc-slot (for IPv4 Flow)

```
user@host> show services accounting flow inline-jflow fpc-slot 5
  Flow information
    FPC Slot: 5
    Flow Packets: 0, Flow Bytes: 0
    Active Flows: 0, Total Flows: 0
    Flows Exported: 0, Flow Packets Exported: 0
    Flows Inactive Timed Out: 0, Flows Active Timed Out: 0
```

show services accounting flow inline-jflow fpc-slot (with IPv4, IPv6, and VPLS Configuration)

```
user@host> show services accounting flow inline-jflow fpc-slot 5
Flow information
    FPC Slot: 5
    Flow Packets: 0, Flow Bytes: 0
    Active Flows: 0, Total Flows: 0
    Flows Exported: 0, Flow Packets Exported: 0
    Flows Inactive Timed Out: 0, Flows Active Timed Out: 0

    IPv4 Flows:
    IPv4 Flow Packets: 0, IPv4 Flow Bytes: 0
    IPv4 Active Flows: 0, IPv4 Total Flows: 0
    IPv4 Flows Exported: 0, IPv4 Flow Packets exported: 0
    IPv4 Flows Inactive Timed Out: 0, IPv4 Flows Active Timed Out: 0

    IPv6 Flows:
    IPv6 Flow Packets: 0, IPv6 Flow Bytes: 0
    IPv6 Active Flows: 0, IPv6 Total Flows: 0
    IPv6 Flows Exported: 0, IPv6 Flow Packets Exported: 0
    IPv6 Flows Inactive Timed Out: 0, IPv6 Flows Active Timed Out: 0

    VPLS Flows:
    VPLS Flow Packets: 0, VPLS Flow Bytes: 0
    VPLS Active Flows: 0, VPLS Total Flows: 0
    VPLS Flows Exported: 0, VPLS Flow Packets Exported: 0
    VPLS Flows Inactive Timed Out: 0, VPLS Flows Active Timed Out: 0
```

show services accounting flow inline-jflow (MX80 Router with IPv4 and IPv6 Configuration)

```
user@host> show services accounting flow inline-jflow
Flow information
    TFEB Slot: 0
    Flow Packets: 0, Flow Bytes: 0
    Active Flows: 0, Total Flows: 0
    Flows Exported: 0, Flow Packets Exported: 0
    Flows Inactive Timed Out: 0, Flows Active Timed Out: 0

    IPv4 Flows:
    IPv4 Flow Packets: 0, IPv4 Flow Bytes: 0
    IPv4 Active Flows: 0, IPv4 Total Flows: 0
    IPv4 Flows Exported: 0, IPv4 Flow Packets exported: 0
    IPv4 Flows Inactive Timed Out: 0, IPv4 Flows Active Timed Out: 0

    IPv6 Flows:
    IPv6 Flow Packets: 0, IPv6 Flow Bytes: 0
    IPv6 Active Flows: 0, IPv6 Total Flows: 0
    IPv6 Flows Exported: 0, IPv6 Flow Packets Exported: 0
    IPv6 Flows Inactive Timed Out: 0, IPv6 Flows Active Timed Out: 0
```

show services accounting flow inline-jflow fpc-slot (PTX1000 Router When Both IPv4 and IPv6 Are Configured)

```
user@host> show services accounting flow inline-jflow fpc-slot 0
Flow information
FPC Slot: 0
Flow Packets: 47427946, Flow Bytes: 5217074060
Active Flows: 0, Total Flows: 2
Flows Exported: 194, Flow Packets Exported: 7045
Flows Inactive Timed Out: 2, Flows Active Timed Out: 192

IPv4 Flows:
IPv4 Flow Packets: 47427946, IPv4 Flow Bytes: 5217074060
IPv4 Active Flows: 0, IPv4 Total Flows: 2
IPv4 Flows Exported: 194, IPv4 Flow Packets exported: 7045
IPv4 Flows Inactive Timed Out: 2, IPv4 Flows Active Timed Out: 192

IPv6 Flows:
IPv6 Flow Packets: 0, IPv6 Flow Bytes: 0
IPv6 Active Flows: 0, IPv6 Total Flows: 0
IPv6 Flows Exported: 0, IPv6 Flow Packets Exported: 0
IPv6 Flows Inactive Timed Out: 0, IPv6 Flows Active Timed Out: 0
```


show services accounting flow-detail

Syntax

show services accounting flow-detail
<detail | extensive | terse>
<*filters*>
<limit *limit-value*>
<name (* | all | *service-name*)>
<order (bytes | packets)>

Release Information Command introduced before Junos OS Release 7.4.

Description Display information about the flows being processed by the accounting service.

Options

none—Display information about all flows.

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

filters—(Optional) Filter the display output of the currently active flow records. The following filters query actively changing data structures and result in different results for multiple invocations:

- **destination-as**—Display flow records filtered by destination autonomous system information.
- **destination-port**—Display flow records filtered by destination port information.
- **destination-prefix**—Display flow records filtered by destination prefix information.
- **input-snmp-interface-index**—Display flow records filtered by SNMP input interface index information.
- **output-snmp-interface-index**—Display flow records filtered by SNMP output interface index information.
- **proto**—Display flow records filtered by protocol type.
- **source-as**—Display flow records filtered by source autonomous system information.
- **source-port**—Display flow records filtered by source port information.
- **source-prefix**—Display flow records filtered by source prefix information.
- **tos**—Display flow records filtered by type of service classification.

limit *limit-value*—(Optional) Limit the display output to the specified number of flows. The default is no limit.

name (* | all | *service-name*)—(Optional) Display information about the flows being processed. Use a wildcard character, specify all services, or provide a specific services name.

order (bytes | packets)—(Optional) Display the flow with the ordering of the highest number, either by byte count or by packet count.

Additional Information When no PIC is active, or when no route record has been downloaded from the PIC, this command reports no flows, even though packets are being sampled. This command displays information about two concurrent sessions only. If a third session is attempted, the command pauses with no output until one of the previous sessions is completed.

Required Privilege Level view

List of Sample Output

show services accounting flow-detail on page 196
show services accounting flow-detail limit on page 197
show services accounting flow-detail name extensive on page 197
show services accounting flow-detail limit order bytes on page 197
show services accounting flow-detail name detail source-port on page 198

Output Fields Table 40 on page 195 lists the output fields for the **show services accounting flow-detail** command. Output fields are listed in the approximate order in which they appear.

Table 40: show services accounting flow-detail Output Fields

Field Name	Field Description	Output Level
Service Accounting interface	Name of the service accounting interface.	All levels
Service name	Name of a service that was configured at the **[edit forwarding-options accounting]** hierarchy level. The default display, **(default sampling)**, indicates the service was configured at the **[edit forwarding-options sampling]** hierarchy level.	All levels
Local interface index	Index counter of the local interface.	All levels
TOS	Type-of-service value from the IP header.	**extensive**
Input SNMP interface index	SNMP index of the interface on which the packet came in.	**extensive**
Output SNMP interface index	SNMP index of the interface on which the packet went out.	**extensive**
Source-AS	Source AS number.	**extensive**
Destination-AS	Destination AS number.	**extensive**
Protocol	Name of the protocol used for the packet flow from the corresponding source address.	All levels
Input interface	Interface on which the packets were received.	All levels
Output interface	Interface on which the packets were transmitted.	All levels
TCP flags	Number of TCP header flags detected in the flow.	**extensive**

Table 40: show services accounting flow-detail Output Fields *(continued)*

Field Name	Field Description	Output Level
Source address	Address where the flow originated.	All levels
Source port	Name of the source port.	All levels
Source prefix length	Source prefix length.	**extensive**
Destination address	Address where the flow is sent.	All levels
Destination prefix length	Destination prefix length.	**extensive**
Destination port	Name of the destination port.	All levels
Start time	Actual time when the packet in this aggregation was first seen.	**detail extensive**
End time	Actual time when the packet in this aggregation was last seen.	**detail extensive**
Packet count	Number of packets in the aggregation.	All levels
Byte count	Number of bytes in the aggregation.	All levels
Time since last active timeout	Amount of time elapsed since the last active timeout, in the format ***hh:mm:ss***.	None specified
Packet count for last active timeout	Number of packets in the aggregation since the last active timeout.	None specified
Byte count for last active timeout	Number of bytes in the aggregation since the last active timeout.	None specified

Sample Output

show services accounting flow-detail

In this sample, the output is split into three sections, with ellipses (...) indicating where the sections are continued.

```
user@host> show services accounting flow-detail
Service Accounting interface: rsp0, Local interface index: 171
Service name: (default sampling)
Interface state: Accounting
Protocol   Input         Source                   Source   Output
           interface     address                    port   interface...
tcp(6)     ge-5/0/1.0    192.0.2.2                     0    ge-5/0/0.0
tcp(6)     ge-5/0/1.0    192.0.2.2                     0    ge-5/0/0.0

Destination          Destination      Packet        Byte  Time since last
address                     port       count       count   active timeout...
198.51.100.149                 0        2660      170240          00:00:58
198.51.100.138                 0        2660      170240          00:00:58
```



```
Packet count for       Byte count for
last active timeout    last active timeout
2805                   179520
2805                   179520
```

show services accounting flow-detail limit

In this sample, the output is split into three sections, with ellipses (...) indicating where the sections are continued.

```
user@host> show services accounting flow-detail limit 1
Service Accounting interface: rsp0, Local interface index: 171
Service name: (default sampling)
Interface state: Accounting
Protocol   Input          Source                   Source   Output
           interface      address                  port     interface...
tcp(6)     ge-5/0/1.0     192.0.2.2                      0   ge-5/0/0.0

Destination          Destination      Packet        Byte    Time since last
address              port             count         count   active timeout...
 198.51.100.149                    0        2158       138112         00:00:47

Packet count for   Byte count for
last active timeout   last active timeout
         2827                  180928
```

show services accounting flow-detail name extensive

```
user@host> show services accounting flow-detail name cf-2 extensive
Service Accounting interface: mo-0/2/0, Local interface index: 145
Service name: cf-2
  TOS: 0, Protocol: udp(17), TCP flags: 0
  Source address: 10.10.10.1, Source prefix length: 0, Destination address:
203.0.113.20,
Destination prefix length: 0, Source port: 1173, Destination port: 69
  Input SNMP interface index: 65, Output SNMP interface index: 0, Source-AS: 0,
Destination-AS: 0
  Start time: 62425, End time: 635265, Packet count: 165845, Byte count: 9453165
```

show services accounting flow-detail limit order bytes

The output of the following command is displayed over 141 columns, not the standard 80 columns. In this sample, the output is split into three sections, with ellipses (...) indicating where the sections are continued.

```
user@host> show services accounting flow-detail limit 5 order bytes
Service Accounting interface: mo-2/0/0, Local interface index: 356
Service name: (default sampling)
           Input          Source                   Source   Output
Protocol   interface      address                  port     interface...
icmp(1)    ge-2/3/0.0     192.0.2.2                      0 .local.
icmp(1)    ge-2/3/0.0     192.0.2.2                      0 .local.
icmp(1)    ge-2/3/0.0     192.0.2.2                      0 .local.
icmp(1)    ge-2/3/0.0     192.0.2.2                      0 .local.
icmp(1)    ge-2/3/0.0     192.0.2.2                      0 .local.

Destination          Destination       Packet       Byte    Time since last
address              port              count        count   active timeout...
192.168.128.2                    0          16        12148   Not applicable
192.168.144.2                    0          16        15229   Not applicable
```

```
192.168.192.2                0          16         13296    Not applicable
192.168.16.2                 0          16         13924    Not applicable
192.168.48.2                 0          16         13428    Not applicable

Packet count for          Byte count for
last active timeout       last active timeout
Not applicable            Not applicable
Not applicable            Not applicable
Not applicable            Not applicable
Not applicable            Not applicable
Not applicable            Not applicable
```

show services accounting flow-detail name detail source-port

```
user@host> show services accounting flow-detail name cf-2 detail source-port 1173
Service Accounting interface: mo-0/2/0, Local interface index: 145
Service name: cf-2
  Protocol: udp(17), Source address: 10.10.10.1, Source port: 1173, Destination
address:
203.0.113.20, Destination port: 69
  Start time: 62425, End time: 811115, Packet count: 142438, Byte count: 8118966
```


show services accounting memory

Syntax show services accounting memory

Release Information Command introduced before Junos OS Release 7.4.

Description Display memory and flow record statistics.

Options This command has no options.

Required Privilege Level view

List of Sample Output show services accounting memory (Monitoring PIC Interface) on page 199
show services accounting memory (Service PIC Interface) on page 200

Output Fields Table 41 on page 199 lists the output fields for the **show services accounting memory** command. Output fields are listed in the approximate order in which they appear.

Table 41: show services accounting memory Output Fields

Output Field	Output Field Description
Service Accounting interface	Name of the service accounting interface.
Memory Utilization	
Local interface index	Index counter of the local interface.
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).

Sample Output

show services accounting memory (Monitoring PIC Interface)

```
user@host> show services accounting memory
Service Accounting interface: mo-2/0/0, Local interface index: 468
  Memory utilization
```



```
    Allocation count: 437340, Free count: 433699, Maximum allocated: 6782
    Allocations per second: 3366, Frees per second: 6412
    Total memory used (in bytes): 133460320,
    Total memory free (in bytes): 133918352
```

show services accounting memory (Service PIC Interface)

```
user@host> show services accounting memory
Service Accounting interface: sp-0/1/0
  Memory utilization
    Allocation count: 1000, Free count: 0
    Allocations per second: 0, Frees per second: 0
    Total memory used (in bytes): 218158272
    Total memory free (in bytes): 587147696

Service Accounting interface: sp-1/0/0
  Memory utilization
    Allocation count: 1000, Free count: 0
    Allocations per second: 0, Frees per second: 0
    Total memory used (in bytes): 218157592
    Total memory free (in bytes): 587148376
```

show services accounting packet-size-distribution

Syntax

show services accounting packet-size-distribution
<name (* | all | *service-name*)>

Release Information

Command introduced before Junos OS Release 7.4.

Description

Display a packet size distribution histogram.

Options

none—Display a packet size distribution histogram of all accounting services.

name (* | all | *service-name*)—(Optional) Display a packet size distribution histogram. Use a wildcard character, specify all services, or provide a specific services name.

Required Privilege Level

view

List of Sample Output

show services accounting packet-size-distribution name on page 201

Output Fields

Table 42 on page 201 lists the output fields for the **show services accounting packet-size-distribution** command. Output fields are listed in the approximate order in which they appear.

Table 42: show services accounting packet-size-distribution Output Fields

Field Name	Field Description
Service Accounting interface	Name of the service accounting interface.
Service name	Name of a service that was configured at the [**edit-forwarding-options accounting**] hierarchy level. The default display, **(default sampling)**, indicates the service was configured at the [**edit-forwarding-options sampling-level**] hierarchy level.
Local interface index	Index counter of the local interface.
Range start	Smallest packet length (in bytes) to count.
Range end	Largest packet length (in bytes) to count.
Number of packets	Count of packets detected in the size between **Range start** and **Range end**.
Percentage packets	Percentage of the total number of packets that are in this size range.

Sample Output

show services accounting packet-size-distribution name

```
user@host> show services accounting packet-size-distribution name test3
Service Accounting interface: mo-0/2/0, Local interface index: 163
Service name: test3
```

```
Range start     Range end     Number of packets     Percentage packets
         32            64                  2924                    100
```

show services accounting status

Syntax

show services accounting status
<inline-jflow fpc-slot *slot-number* | name (* | all | *service-name*)>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 13.2R2 for EX Series switches.

Description

Display available Physical Interface Cards (PICs) for accounting services.

Options

none—Display available PICs for all accounting services.

inline-jflow fpc-slot *slot-number*—(Optional) Display inline flow accounting status for the specified FPC. For a two-member MX Series Virtual Chassis or EX9200 Virtual Chassis, the master router or switch uses FPC slot numbers 0 through 11 with no offset; the backup router or switch uses FPC slot numbers 12 through 23, with an offset of 12.

name (* | all | *service-name*)—(Optional) Display available PICs. Use a wildcard character, specify all services, or provide a specific services name.

Required Privilege Level

view

Related Documentation

- show services accounting flow on page 189
- *Inline Flow Monitoring for Virtual Chassis Overview*

List of Sample Output

show services accounting status name (Monitoring PIC Interface) on page 204
show services accounting status name (Service PIC Interface) on page 205
show services accounting status inline-jflow fpc-slot (When Both IPv4 and IPv6 Are Configured) on page 205
show services accounting status inline-jflow (MX80 Router When Both IPv4 and IPv6 Are Configured) on page 205
show services accounting status inline-jflow fpc-slot (PTX1000 Router When Both IPv4 and IPv6 Are Configured) on page 205

Output Fields

Table 43 on page 203 lists the output fields for the **show services accounting status** command. Output fields are listed in the approximate order in which they appear.

Table 43: show services accounting status Output Fields

Field	Field Description
Service Accounting interface	Name of the service accounting interface.
Service name	Name of a service that was configured at the [**edit-forwarding-options accounting**] hierarchy level. The default display,**(default sampling)**, indicates the service was configured at the [**edit-forwarding-options sampling-level**] hierarchy level.

Table 43: show services accounting status Output Fields *(continued)*

Field	Field Description
FPC Slot	Slot number of the FPC for which the flow information is displayed.
Local interface index	Index counter of the local interface.
Interface state	Accounting state of the passive monitoring interface. • **Accounting**—PIC is actively accounting. • **Disabled**—PIC has been disabled from the CLI. • **Not accounting**—PIC is up but not accounting. This can happen while the PIC is coming online, or when the PIC is up but has no logical unit configured under the physical interface. • Unknown
Group index	Integer that represents the monitoring group of which the PIC is a member. **Group index** is a mapping from the group name to an index. It is not related to the number of monitoring groups.
Export interval (in seconds)	Configured export interval for cflowd records, in seconds.
Export format	Configured export format.
Protocol	Protocol the PIC is configured to monitor.
Engine type	Configured engine type that is inserted in output cflowd packets.
Engine ID	Configured engine ID that is inserted in output cflowd packets.
Route Record Count	Number of routes recorded.
AS Record Count	Number of autonomous systems recorded.
Route Records Set	Status of route recording; whether routes are recorded or not.
Configuration Set	Status of monitoring configuration; whether monitoring configuration is set or not.

Sample Output

show services accounting status name (Monitoring PIC Interface)

```
user@host> show services accounting status name count1
Service Accounting interface: mo-2/0/0, Local interface index: 468
Service name: count1
Interface state: Accounting
  Group index: 0
  Export interval (in seconds): 60, Export format: cflowd v8
  Protocol: IPv4, Engine type: 55, Engine ID: 5
```


Sample Output

show services accounting status name (Service PIC Interface)

```
user@host> show services accounting status name
Service Accounting interface: sp-0/1/0
Interface state: Accounting
  Export format: 9, Route record count: 0
  IFL to SNMP index count: 7, AS count: 0
  Configuration set: Yes, Route record set: No, IFL SNMP map set: Yes

Service Accounting interface: sp-1/0/0
Interface state: Accounting
  Export format: 9, Route record count: 33
  IFL to SNMP index count: 7, AS count: 1
  Configuration set: Yes, Route record set: Yes, IFL SNMP map set: Yes
```

show services accounting status inline-jflow fpc-slot (When Both IPv4 and IPv6 Are Configured)

```
user@host> show services accounting status inline-jflow fpc-slot 5
FPC Slot: 5
    IPV4 export format: Version-IPFIX, IPV6 export format: Version-IPFIX
    VPLS export format: Not set
    IPv4 Route Record Count: 5, IPv6 Route Record Count: 7
    Route Record Count: 12, AS Record Count: 1
    Route-Records Set: Yes, Config Set: Yes
```

show services accounting status inline-jflow (MX80 Router When Both IPv4 and IPv6 Are Configured)

```
user@host> show services accounting status inline-jflow

Status information
    TFEB Slot: 0
    Export format: IP-FIX
    IPv4 Route Record Count: 6, IPv6 Route Record Count: 8
    Route Record Count: 14, AS Record Count: 1
    Route-Records Set: Yes, Config Set: Yes
```

show services accounting status inline-jflow fpc-slot (PTX1000 Router When Both IPv4 and IPv6 Are Configured)

```
user@host> show services accounting status inline-jflow fpc-slot 0
Status information
FPC Slot: 0
IPV4 export format: Version-IPFIX, IPV6 export format: Version-IPFIX
VPLS export format: Not set, MPLS export format: Not set
IPv4 Route Record Count: 23, IPv6 Route Record Count: 3, MPLS Route Record Count:
 0
Route Record Count: 26, AS Record Count: 1
Route-Records Set: Yes, Config Set: Yes
IPv4 MAX FLOW Count: 0, IPv6 MAX FLOW Count: 0
VPLS MAX FLOW Count: 0, MPLS MAX FLOW Count: 2
```

show services accounting usage

Syntax

show services accounting usage
<name *service-name*>

Release Information Command introduced before Junos OS Release 7.4.

Description Display the CPU usage of PIC used for active flow monitoring.

Options

none—Display CPU usage for all service names.

name *service-name*—(Optional) Display CPU usage for the specified service name.

Additional Information When no route record has been downloaded from the PIC, this command reports no flows, even though packets are being sampled.

Required Privilege Level view

List of Sample Output

show services accounting usage (Monitoring PIC Interface) on page 207
show services accounting usage (Service PIC Interface) on page 207

Output Fields Table 44 on page 206 lists the output fields for the **show services accounting usage** command. Output fields are listed in the approximate order in which they appear.

Table 44: show services accounting usage Output Fields

Output Field	Output Field Description
Service Accounting interface	Name of the service accounting interface.
Service name	Name of a service that was configured at the [**edit-forwarding-options accounting**] hierarchy level. The default display, **(default sampling)**, indicates the service was configured at the [**edit-forwarding-options sampling-level**] hierarchy level.
Local interface index	Index counter of the local interface.
Uptime	Time that the PIC has been operational (in milliseconds).
Interrupt time	Total time that the PIC has spent processing packets since the last PIC reset (in microseconds).
Load (5 second)	CPU load on the PIC, averaged more than 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 more than 1 minute. The number is a percentage obtained by dividing the time spent on active tasks by the total elapsed time.

Sample Output

show services accounting usage (Monitoring PIC Interface)

```
user@host> show services accounting usage
Service Accounting interface: mo-1/1/0, Local interface index: 15
Service name: (default sampling)
  CPU utilization
    Uptime: 600413856 milliseconds, Interrupt time: 2403 microseconds
    Load (5 second): 43%, Load (1 minute): 24%
```

show services accounting usage (Service PIC Interface)

```
user@host> show services accounting usage
Service Accounting interface: sp-0/1/0
Service name: (default sampling)
  CPU utilization
    Uptime: 7853940 milliseconds, Interrupt time: 0 microseconds
    Load (5 second): 2%, Load (1 minute): 0%


Service Accounting interface: sp-0/1/0
Service name: (default sampling)
  CPU utilization
    Uptime: 331160 milliseconds, Interrupt time: 0 microseconds
    Load (5 second): 2%, Load (1 minute): 0%
```


CHAPTER 10

Dynamic Flow Capture Commands

clear services dynamic-flow-capture

Syntax

clear services dynamic-flow-capture capture-group *group-name*
<criteria-identifier *identifier*>
<destination-identifier *identifier*>
<force>
<static>

Release Information Command introduced in Junos OS Release 7.4.

Description (M320 Series routers and T Series routers only) Clear dynamic flow capture information for specified capture group.

Options

capture-group *group-name*—Use the specified capture-group identifier.

criteria-identifier *identifier*—(Optional) Use the specified criteria identifier.

destination-identifier *identifier*—(Optional) Use the specified content destination identifier.

force—(Optional) Force clearing of criteria.

static—(Optional) Clear static criteria.

Required Privilege Level network

List of Sample Output clear services dynamic-flow-capture on page 210

Output Fields When you enter this command, you are not provided feedback on the status of your request.

Sample Output

clear services dynamic-flow-capture

```
user@host> clear services dynamic-flow-capture capture-group flow-a
```


show services dynamic-flow-capture content-destination

Syntax

show services dynamic-flow-capture content-destination capture-group *group-name*
  destination-identifier *identifier*
<terse>

Release Information

Command introduced in Junos OS Release 7.4.

Description

(M320 Series routers and T Series routers only) Display information about the content destination that receives packets from the dynamic flow capture (DFC) interface.

Options

capture-group ***group-name***—Display information for the specified capture-group identifier.

destination-identifier ***identifier***—Display information for the specified content destination identifier.

terse—(Optional) Display summary information.

Required Privilege Level

view

List of Sample Output

show services dynamic-flow-capture content-destination capture-group on page 212

Output Fields

Table 45 on page 211 lists the output fields for the **show services dynamic-flow-capture content-destination** command. Output fields are listed in the approximate order in which they appear.

Table 45: show services dynamic-flow-capture content-destination Output Fields

Output Field	Output Field Description
Capture group	Name of the capture group.
Content destination	Name of the content destination.
Criteria	Number of criteria specified.
Bandwidth	Bandwidth used by the matched traffic.
Matched packets	Number of matched packets sent to the content destination.
Matched bytes	Number of matched bytes sent to the content destination.
Congestion notifications	Number of notification messages sent.

Sample Output

show services dynamic-flow-capture content-destination capture-group

```
user@host> show services dynamic-flow-capture content-destination capture-group g1
destination-identifier cd1 terse
  Capture group: g1, Content destination: cd1, Criteria: 0, Bandwidth: 0, Matched
 packets: 0, Matched bytes: 0, Congestion notifications: 0
```


show services dynamic-flow-capture control-source

Syntax

show services dynamic-flow-capture control-source capture-group *group-name*
  control-source source-identifier *identifier*
<detail | terse>

Release Information

Command introduced in Junos OS Release 7.4.

Description

(M320 Series routers and T Series routers only) Display information about the control source that makes dynamic flow capture requests to the dynamic flow capture interface.

Options

capture-group *group-name*—Capture group identifier.

source-identifier *identifier*—Control source identifier.

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

Required Privilege Level

view

List of Sample Output

show services dynamic-flow-capture control-source source-identifier capture-group on page 214
show services dynamic-flow-capture control-source ource-identifier capture-group detail on page 214

Output Fields

Table 46 on page 213 lists the output fields for the **show services dynamic-flow-capture control-source** scommand. Output fields are listed in the approximate order in which they appear.

Table 46: show services dynamic-flow-capture control-source Output Fields

Output Field	Output Field Description
Capture group	Name of the capture group.
Control source	Name of the control source.
Criteria added, Criteria add failed	Number of criteria added or added and failed.
Active criteria	Number of active criteria.
Static criteria, Dynamic criteria	Number of static or dynamic criteria.
Control protocol requests	Total number of control protocol requests.
Requests	Number of **Add**, **Delete**, **List**, **Refresh**, and **No-op** control protocol requests.
Failed	Number of **Add**, **Delete**, **List**, **Refresh**, and **No-op** failed control protocol requests.

Table 46: show services dynamic-flow-capture control-source Output Fields *(continued)*

Output Field	Output Field Description
Add request rate	Rate of add requests.
Add request peak rate	Peak rate of add requests.
Bandwidth across all criteria	Bandwidth used by all the requests.
Total notifications	Total number of notifications sent and the number of notifications by category: **Restart**, **Rollover**, **Timeout**, **Congestion**, **Congestion delete**, and **Dups** (duplicates) dropped.
Criteria deleted	Total number of criteria deleted and the number of deleted criteria by category: **Timeout idle**, **Timeout total**, **Packets**, and **Bytes**.
Sequence number	Sequence number.

Sample Output

show services dynamic-flow-capture control-source source-identifier capture-group

```
user@host> show services dynamic-flow-capture control-source source-identifier cs0_cg0
capture-group cg_0
Capture group: cg_0, Control source: cs0_cg0
 Criteria added: 28, Criteria add failed: 0, Active criteria: 0, Control protocol
 requests: 28,  Add request rate: 0,
 Add request peak rate: 1, Bandwidth across all criteria: 0, Total notifications:
 1, Criteria deleted: 28, Sequence number: 0
```

show services dynamic-flow-capture control-source ource-identifier capture-group detail

```
user@host> show services dynamic-flow-capture control-source source-identifier cs0_cg0
capture-group cg_0 detail
Capture group: cg_0, Control source: cs0_cg0
  Criteria added: 28, Criteria add failed: 0
  Active criteria: 0
    Static criteria: 0, Dynamic criteria: 0
  Control protocol requests: 28
                     Add        Delete         List       Refresh         No-op

    Requests          28             0            0             0             0

    Failed             0             0            0             0             0

  Add request rate: 0
  Add request peak rate: 1
  Bandwidth across all criteria: 0
  Total notifications: 1
    Restart: 1, Rollover: 0, No-op: 0, Timeout: 0, Congestion: 0, Congestion
delete: 0, Dups dropped: 0
  Criteria deleted: 28
    Timeout idle: 0, Timeout total: 0, Packets: 0, Bytes: 0
  Sequence number: 0
```

show services dynamic-flow-capture statistics

Syntax show services dynamic-flow-capture statistics capture-group *group-name*

Release Information Command introduced in Junos OS Release 7.4.

Description (M320 Series routers and T Series routers only) Display statistics information about the capture group specified for dynamic flow capture.

Options **capture-group** ***group-name***—Display information for the specified capture group identifier.

Required Privilege Level view

List of Sample Output show services dynamic-flow-capture statistics capture-group on page 217

Output Fields Table 47 on page 216 lists the output fields for the **show services dynamic-flow-capture statistics** command. Output fields are listed in the approximate order in which they appear.

Table 47: show services dynamic-flow-capture statistics Output Fields

Output Field	Output Field Description
Input	Incoming dynamic flow capture packet statistics: • **Control protocol packets**—Number of control protocol packets received. • **Captured data packets**—Number of data packets captured. • **Control IRI packets**—Number of control IRI packets received.
Control protocol drops	Control protocol packets dropped for the following reasons: • **Not IP packets**—Dropped packets were not IP packets. • **Not UDP packets**—Dropped packets were not User Datagram Protocol (UDP) packets. • **Invalid destination address**—Dropped packets had invalid destination addresses. • **No memory**—Packets dropped because of insufficient memory. • **Unauthorized control source**—Packets dropped because the control source was not authenticated. • **Bad request**—Packets dropped because the request was invalid. • **Unknown control source**—Packets dropped because the control source was not known. • **Not DTCP**—Dropped packets did not adhere to the control protocol format. • **Bad command line**—Packets dropped because of a version mismatch. • **Bandwidth exceeded**—Packets dropped because the bandwidth was exceeded. • **Drop rate due to exceeded bandwidth**—Rate of traffic dropped because the bandwidth was exceeded. • **Other**—Packets dropped for other reasons or undetermined causes.

Table 47: show services dynamic-flow-capture statistics Output Fields *(continued)*

Output Field	Output Field Description
Input drops	Incoming dynamic flow capture packets dropped for the following reasons: • **Unknown packets**—Packets dropped because the packet type was not recognized. • **Captured data not IPv4**—Packets dropped because they were not IPv4 packets. • **Captured data too small**—Packets dropped because they were smaller than the size reported in their headers. • **Captured data drops**—Data packets dropped because of undetermined causes. • **Captured data not matched**—Packets dropped because they did not match filter criteria. • **Bandwidth exceeded**—Packets dropped because the bandwidth was exceeded. • **Drop rate due to exceeded bandwidth**—Rate of traffic dropped because the bandwidth was exceeded.
Output	Outgoing dynamic flow capture packet statistics: • **Control protocol packets**—Number of control protocol packets sent. • **Captured data packets**—Number of captured data packets sent.
Output drops	Outgoing packets dropped: • **Control protocol drops**—Number of control protocol packets dropped. • **Captured data drops**—Number of captured data packets dropped.
Flow Statistics	DFC flow statistics: • **Active flow cache entries** • **Active flow cache usage percentage** • **Flow cache entries allocated** • **Number of control sources** • **Number of content destinations** • **Number of criteria** • **Maximum criteria matching one flow** • **Cached flows purged for memory** • **Maximum filters matching one packet**

Sample Output

show services dynamic-flow-capture statistics capture-group

```
user@host> show services dynamic-flow-capture statistics capture-group g1
Input:

  Control protocol packets: 643, Captured data packets: 69977, Control IRI packets:
 337

Control protocol drops:

  Not IP packets: 0, Not UDP packets: 3, Invalid destination address: 0, No memory:
 0, Unauthorized control source: 0,

  Bad request: 0, Unknown control source: 0, Not DTCP: 0, Bad command line: 0,
Bandwidth exceeded: 0,
```

```
  Drop rate due to exceeded bandwidth: 0, Other: 0

Input drops:

  Unknown packets: 0, Captured data not IPv4: 0, Captured data too small: 0,
Captured data drops: 0, Captured data not matched: 0,

  Bandwidth exceeded: 0, Drop rate due to exceeded bandwidth: 0

Output:

  Control protocol packets: 644, Captured data packets: 1119624

Output drops:

  Control protocol drops: 0, Captured data drops: 0

Flow Statistics:

  Active flow cache entries: 40, Active flow cache usage percentage: 0, Flow cache
 entries allocated: 40,

  Number of control sources: 4, Number of content destinations: 64, Number of
criteria: 640,

  Maximum criteria matching one flow: 16, Cached flows purged for memory: 0,
Maximum filters matching one packet: 16
```


CHAPTER 11

Flow Collection Commands

clear services flow-collector statistics

Syntax clear services flow-collector statistics (all | interface *interface-name*)

Release Information Command introduced before Junos OS Release 7.4.

Description (M40e, M160, and M320 Series routers and T Series routers only) Clear statistics for one flow collector interface or for all flow collector interfaces.

Options **all**—Clear statistics for all configured flow collector interfaces.

interface *interface-name*—Clear statistics for the specified flow collector interface (**cp-*fpc/pic/port***).

Required Privilege Level network

List of Sample Output clear services flow-collector statistics on page 220

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear services flow-collector statistics

```
user@host> clear services flow-collector statistics interface cp-5/0/0
Flow collector interface: cp-5/0/0
Interface state: Collecting flows
Statistics cleared successfully
```


request services flow-collector change-destination primary interface

Syntax

request services flow-collector change-destination primary interface cp-*fpc/pic/port*
<clear-files>
<clear-logs>
<immediately | gracefully>

Release Information

Command introduced before Junos OS Release 7.4.

Description

(M40e, M160, and M320 Series routers and T Series routers only) Switch to the primary File Transfer Protocol (FTP) server that is configured as a flow collector.

Options

none—Switch to the primary FTP server.

cp-*fpc/pic/port*—Use the specified flow collector interface name for the primary destination.

clear-files—(Optional) Request clearing of existing data files in the FTP wait queue when the switch takes place.

clear-logs—(Optional) Request clearing of existing logs when the switch takes place.

immediately | gracefully—(Optional) Specify whether you want the switch to take place immediately, or to affect only newly created files.

Required Privilege Level

maintenance

List of Sample Output

request services flow-collector change-destination primary interface on page 221

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request services flow-collector change-destination primary interface

```
user@host> request services flow-collector change-destination primary interface cp-6/0/0
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Destination change successful
```


request services flow-collector change-destination secondary interface

Syntax

request services flow-collector change-destination secondary interface cp-*fpc/pic/port*
<clear-files>
<clear-logs>
<immediately | gracefully>

Release Information Command introduced before Junos OS Release 7.4.

Description (M40e, M160, and M320 Series routers and T Series routers only) Switch to the secondary File Transfer Protocol (FTP) server that is configured as a flow collector.

Options

none—Switch to the secondary FTP server.

cp-*fpc/pic/port*—Use the specified flow collector interface name (**cp-*fpc/pic/port***) for the secondary destination.

clear-files—(Optional) Request clearing of existing data files in the FTP wait queue when the switch takes place.

clear-logs—(Optional) Request clearing of existing logs when the switch takes place.

immediately | gracefully—(Optional) Specify whether you want the switch to take place immediately, or to affect only newly created files.

Required Privilege Level maintenance

List of Sample Output request services flow-collector change-destination secondary interface on page 222

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request services flow-collector change-destination secondary interface

```
user@host> request services flow-collector change-destination secondary interface cp-6/0/0
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Destination change successful
```


request services flow-collector test-file-transfer

Syntax request services flow-collector test-file-transfer *filename* interface (all | cp-*fpc/pic/port*) (channel-zero | channel-one) (primary | secondary)

Release Information Command introduced before Junos OS Release 7.4.

Description (M40e, M160, and M320 Series routers, PTX Series, and T Series routers only) Transfer a test file to the primary or secondary File Transfer Protocol (FTP) server that is configured as a flow collector. This command verifies that the output side of the flow collector interface is operating properly.

Options *filename*—Name of the test file to transfer.

interface (all | cp-*fpc/pic/port*)—Transfer a test file of flows from all configured flow collector interfaces or from only the specified interface.

channel-zero | channel-one—Transfer a file from export channel 0 (unit 0) or channel 1 (unit 1) of the PIC.

primary | secondary—Transfer a file to the primary or secondary server configured as a flow collector.

Required Privilege Level network

List of Sample Output request services flow-collector test-file-transfer interface channel-one primary on page 223

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request services flow-collector test-file-transfer interface channel-one primary

```
user@host> request services flow-collector test-file-transfer test_file interface cp-7/1/0
channel-one primary

Flow collector interface: cp-7/1/0
Interface state: Collecting flows
Response: Test file transfer successfully scheduled
```

show services flow-collector file interface

Syntax show services flow-collector file interface (all | cp-*fpc/pic/port*)
<detail | extensive | terse>

Release Information Command introduced before Junos OS Release 7.4.

Description (M40e, M160, and M320 Series routers and T Series routers only) Display information about flow collector files.

Options **none**—Display file information for all configured flow collector interfaces.

all | cp-*fpc/pic/port*—Display file information for all configured flow collector interfaces or for the specified interface.

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

Additional Information No entries are displayed for files that have been successfully transferred.

Required Privilege Level view

List of Sample Output show services flow-collector file interface extensive on page 225

Output Fields Table 48 on page 224 lists the output fields for the **show services flow-collector file interface** command. Output fields are listed in the approximate order in which they appear.

Table 48: show services flow-collector file interface Output Fields

Output Field	Output Field Description	Level of Output
Filename	Name of the file created on the flow collector interface.	All levels
Flows	Total number of collector flows for which records are present in the file.	none specified

Throughput	Throughput statistics:
	• **Flow records**—Number of flow records in the file.
	  ◦ **per second**—Average number of flow records per second.
	  ◦ **peak per second**—Peak number of flow records per second.
	• **Uncompressed bytes**—Total file size before compression.
	  ◦ **per second**—Average number of uncompressed bytes per second.
	  ◦ **peak per second**—Peak number of uncompressed bytes per second.
	• **Compressed bytes**—Total file size after compression.
	  ◦ **per second**—Average number of compressed bytes per second.
	  ◦ **peak per second**—Peak number of compressed bytes per second.
		extensive

Table 48: show services flow-collector file interface Output Fields *(continued)*

Output Field	Output Field Description	Level of Output

Status	File statistics:
	• **Compressed blocks**—(**extensive** output only) Data blocks in the file that have been compressed. The file is exported only when the compressed block count and block count become the same.
	• **Block count**—(**extensive** output only) Total number of data blocks in the file.
	• **State**—Processing state of the file.
	  ◦ **Active**—The flow collector interface is writing to the file.
	  ◦ **Export 1**—File export is in progress to the primary server.
	  ◦ **Export 2**—File export is in progress to the secondary server.
	  ◦ **Wait**—File is pending export.
	  ◦ **Transfer attempts 0**.—Number of attempts made to transfer the file. If the file is successfully transferred in the first attempt, this field is 0.
		All levels

Sample Output

show services flow-collector file interface extensive

```
user@host> show services flow-collector file interface cp-3/2/0 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
```


show services flow-collector input interface

Syntax

show services flow-collector input interface (all | cp-*fpc/pic/port*)
<detail | extensive | terse>

Release Information

Command introduced before Junos OS Release 7.4.

Description

(M40e, M160, and M320 Series routers and T Series routers only) Display the number of packets received by collector interfaces from monitoring interfaces.

Options

none—Display packets received by all configured flow collector interfaces.

all | cp-*fpc/pic/port*—Display packets received by all configured flow collector interfaces or by the specified interface.

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

Required Privilege Level

view

List of Sample Output

show services flow-collector input interface on page 226
show services flow-collector input interface all on page 226

Output Fields

Table 49 on page 226 lists the output fields for the **show services flow-collector input interface** command. Output fields are listed in the approximate order in which they appear.

Table 49: show services flow-collector input interface Output Fields

Output Field	Output Field Description
Interface	Name of the monitoring interface.
Packets	Number of packets traveling from the monitoring interface to the flow collector interface.
Bytes	Number of bytes traveling from the monitoring interface to the flow collector interface.

Sample Output

show services flow-collector input interface

```
user@host> show services flow-collector input interface cp-3/2/0
Interface                       Packets        Bytes
mo-3/0/0.0                        21706     32328568
mo-3/1/0.0                        21706     32329096
```

show services flow-collector input interface all

```
user@host> show services flow-collector input interface all
Flow collector interface: cp-6/1/0
Interface state: Collecting flows
Interface                       Packets        Bytes
```

```
mo-3/0/0.0                          274       416232
mo-3/3/0.0                          274       416184
mo-1/0/0.0                          274       416232
mo-1/1/0.0                          274       416232
mo-1/2/0.0                          274       416232
mo-1/3/0.0                          274       416232
mo-3/1/0.0                          274       416232
mo-4/0/0.0                          274       416232
mo-4/1/0.0                          274       416232
mo-4/2/0.0                          274       416184
mo-4/3/0.0                          274       416232
mo-5/0/0.0                          274       416232
mo-5/1/0.0                          274       416232
mo-5/2/0.0                          274       416232
mo-5/3/0.0                          274       416232
mo-6/0/0.0                          274       416232

Flow collector interface: cp-6/3/0
Interface state: Collecting flows
```

show services flow-collector interface

Syntax

show services flow-collector interface (all | cp-*fpc/pic/port*)
<detail | extensive | terse>

Release Information

Command introduced before Junos OS Release 7.4.

Description

(M40e, M160, and M320 Series routers and T Series routers only) Display overall statistics for the flow collector application.

Options

none—Display statistics for flow collector applications on all interfaces.

all | cp-*fpc/pic/port*—Display statistics for flow collector applications on all interfaces or for the specified interface.

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

Required Privilege Level

view

List of Sample Output

show services flow-collector interface all detail on page 230
show services flow-collector interface all extensive on page 231
show services flow-collector interface all terse on page 233
show services flow-collector interface extensive on page 233

Output Fields

Table 50 on page 228 lists the output fields for the **show services flow-collector interface** command. Output fields are listed in the approximate order in which they appear.

Table 50: show services flow-collector interface Output Fields

Output Field	Output Field Description	Level of Output
Flow collector interface	Name of the flow collector interface.	All levels
Interface state	Collecting flow state for the interface.	All levels
Packets	Total number of packets received.	none specified
Flows Uncompressed Bytes	Total uncompressed data size for all files created on this PIC.	none specified
Compressed Bytes	Total compressed data size for all files created on this PIC.	none specified
FTP bytes	Total number of bytes transferred to the FTP server, including those dropped during transfer.	none specified
FTP files	Total number of FTP transfers attempted by the server.	none specified
Memory	Bytes used on the PIC and bytes free.	**detail extensive**

Table 50: show services flow-collector interface Output Fields *(continued)*

Output Field	Output Field Description	Level of Output

Input	Incoming flow collector packet statistics:
	• **Packets**—Number of packets received on the unit.
	  ◦ **per second**—Average number of packets per second.
	  ◦ **peak per second**—Peak number of packets per second.
	• **Bytes**—Number of bytes received on the unit.
	  ◦ **per second**—Average number of bytes per second.
	  ◦ **peak per second**—Peak number of bytes per second.
	• **Flow records processed**—Number of records in the flow collector packets that were processed by the flow-collector interface.
	  ◦ **per second**—Average number of flow records processed per second.
	  ◦ **peak per second**—Peak number of flow records per second.
		detail extensive

Allocation	Data block statistics:
	• **Blocks allocated**—Total number of data blocks (containing flow records) allocated to the files created on this PIC.
	  ◦ **per second**—Average number of blocks allocated per second.
	  ◦ **peak per second**—Peak number of blocks allocated per second.
	• **Blocks freed**—Total number of data blocks freed.
	  ◦ per second—Average number of blocks freed per second.
	  ◦ peak per second—Peak number of blocks freed per second.
	• **Blocks unavailable**—Total number of data block requests denied, typically because of a memory shortage.
	  ◦ per second—Average number of blocks unavailable per second.
	  ◦ peak per second—Peak number of blocks unavailable per second.
		extensive

Files	File statistics, incremented since the PIC last booted: • **Files created**—Total number of files created on this PIC. • **Files exported**— Number of files successfully created and exported. • **Files destroyed**—(**extensive** output only) Number of files successfully exported and files dropped by the flow collection interface.	**detail extensive**

Throughput	Throughput statistics:
	• **Uncompressed bytes**—Total uncompressed data size for all files created on this PIC.
	  ◦ **per second**—Average number of uncompressed bytes per second.
	  ◦ **peak per second**—Peak number of uncompressed bytes per second.
	• **Compressed bytes**—Total compressed data size for all files created on this PIC.
	  ◦ **per second**—Average number of compressed bytes per second.
	  ◦ **peak per second**—Peak number of compressed bytes per second.
		detail extensive

Table 50: show services flow-collector interface Output Fields *(continued)*

Output Field	Output Field Description	Level of Output
Packet drops	Number of packets dropped for the following causes: • **No memory**—Packets dropped because of insufficient memory. • **Not IP**—Packets dropped because they are not IP packets. • **Not IPv4**—Packets dropped because they are not IP version 4 packets. • **Too small**—Packets dropped because each packet was smaller than the size reported in its header. • **Fragments**—Packets dropped because of fragmentation. Fragments are not reassembled. • **ICMP**—Packets dropped because they are not ICMP packets. • **TCP**—Packets dropped because they are not TCP packets. • **Unknown**—Packets dropped because of undetermined causes. • **Not Junos flow**—Packets dropped because they are not interpreted by Junos OS. Junos OS interprets only IPv4, UDP cflowd version 5 packets.	**extensive**
File transfer	File transfer statistics: • **FTP bytes**—Total number of bytes transferred to the FTP server, including those dropped during transfer. • **FTP files**—Total number of FTP transfers attempted by the server. • **FTP failure**—Total number of FTP failures encountered by the server.	**detail extensive**
Flow collector interface	Physical interface acting as a flow collector.	**detail**

Export channel	Export channel **0** is unit 0. Export channel **1** is unit 1. Flow receive channel is unit 2. Server status statistics are the following:
	• **Current server Primary** or **Secondary**—Current FTP server being used. Value is
	• **Primary server state**—State of the server:
	  ◦ **OK**—Server is operating without problems.
	  ◦ **FTP error**—Server encountered an FTP protocol error while sending files.
	  ◦ **Network error**—Flow-collector interface has errors when contacting the primary FTP server.
	  ◦ **Unknown**—First file transfer has not been sent to the primary server.
	• **Secondary server state**—State of the server:
	  ◦ **OK**—Server is operating without errors.
	  ◦ **FTP error**—Server encountered an FTP protocol error while sending files.
	  ◦ **Network error**—Flow-collector interface has errors when contacting the secondary FTP server.
	  ◦ **Unknown**—First file transfer has not been sent to the secondary server.
	• **Not configured**—Secondary server is not configured.
		detail extensive

Sample Output

show services flow-collector interface all detail

`user@host>` **show services flow-collector interface all detail**

```
Flow collector interface: cp-6/1/0
Interface state: Collecting flows
Memory:
    Used: 51452732, Free: 440329088
Input:
    Packets: 4384, per second: 0, peak per second: 156
    Bytes: 6659616, per second: 0, peak per second: 249695
    Flow records processed: 131070, per second: 0, peak per second: 4914
Files:
    Files created: 1, per second: 0, peak per second: 0
    Files exported: 1, per second: 0, peak per second: 0
Throughput:
    Uncompressed bytes: 13742307, per second: 0, peak per second: 593564
    Compressed bytes: 3786177, per second: 0, peak per second: 162826
File Transfer:
    FTP bytes: 3786247, per second: 0, peak per second: 378620
    FTP files: 1, per second: 0, peak per second: 0
    FTP failure: 0
Export channel: 0
    Current server: Primary
    Primary server state: OK, Secondary server state: OK
Export channel: 1
    Current server: Primary
    Primary server state: Unknown, Secondary server state: OK

Flow collector interface: cp-6/3/0
Interface state: Collecting flows
Memory:
    Used: 51452732, Free: 440329088
Input:
    Packets: 0, per second: 0, peak per second: 0
    Bytes: 0, per second: 0, peak per second: 0
    Flow records processed: 0, per second: 0, peak per second: 0
Files:
    Files created: 0, per second: 0, peak per second: 0
    Files exported: 0, per second: 0, peak per second: 0
Throughput:
    Uncompressed bytes: 0, per second: 0, peak per second: 0
    Compressed bytes: 0, per second: 0, peak per second: 0
File Transfer:
    FTP bytes: 70, per second: 0, peak per second: 6
    FTP files: 0, per second: 0, peak per second: 0
    FTP failure: 0
Export channel: 0
    Current server: Primary
    Primary server state: Unknown, Secondary server state: OK
Export channel: 1
    Current server: Primary
    Primary server state: Unknown, Secondary server state: OK
```

show services flow-collector interface all extensive

```
user@host> show services flow-collector interface all extensive
Flow collector interface: cp-6/1/0
Interface state: Collecting flows
Memory:
    Used: 51452732, Free: 440329088
Input:
    Packets: 4384, per second: 0, peak per second: 156
    Bytes: 6659616, per second: 0, peak per second: 249695
    Flow records processed: 131070, per second: 0, peak per second: 4914
```

```
Allocation:
    Blocks allocated: 108, per second: 0, peak per second: 0
    Blocks freed: 108, per second: 0, peak per second: 10
    Blocks unavailable: 0, per second: 0, peak per second: 0
Files:
    Files created: 1, per second: 0, peak per second: 0
    Files exported: 1, per second: 0, peak per second: 0
    Files destroyed: 1, per second: 0, peak per second: 0
Throughput:
    Uncompressed bytes: 13742307, per second: 0, peak per second: 593564
    Compressed bytes: 3786177, per second: 0, peak per second: 162826
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: 3786247, per second: 0, peak per second: 378620
    FTP files: 1, per second: 0, peak per second: 0
    FTP failure: 0
Export channel: 0
    Current server: Primary
    Primary server state: OK, Secondary server state: OK
Export channel: 1
    Current server: Primary
    Primary server state: Unknown, Secondary server state: OK

Flow collector interface: cp-6/3/0
Interface state: Collecting flows
Memory:
    Used: 51452732, Free: 440329088
Input:
    Packets: 0, per second: 0, peak per second: 0
    Bytes: 0, per second: 0, peak per second: 0
    Flow records processed: 0, per second: 0, peak per second: 0
Allocation:
    Blocks allocated: 0, 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: 0, 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: 0, per second: 0, peak per second: 0
    Compressed bytes: 0, per second: 0, peak per second: 0
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: 70, per second: 0, peak per second: 6
    FTP files: 0, per second: 0, peak per second: 0
    FTP failure: 0
Export channel: 0
    Current server: Primary
    Primary server state: Unknown, Secondary server state: OK
Export channel: 1
```



```
    Current server: Primary
    Primary server state: Unknown, Secondary server state: OK
```

show services flow-collector interface all terse

```
user@host> show services flow-collector interface all terse
Flow collector interface: cp-6/1/0
Interface state: Collecting flows
  Packets      Bytes      Flows Uncompressed  Compressed     FTP bytes FTP files
                                       Bytes       Bytes
     4384    6659616     131070     13742307     3786177       3786247         1

Flow collector interface: cp-6/3/0
Interface state: Collecting flows
  Packets      Bytes      Flows Uncompressed  Compressed     FTP bytes FTP files
                                       Bytes       Bytes
        0          0          0            0           0            70         0
```

show services flow-collector interface extensive

```
user@host> show services flow-collector interface cp-5/2/0 extensive
Flow collector interface: cp-5/2/0
Interface state: Collecting flows
Memory:
     Used: 458311860, Free: 40810008
Input:
     Packets: 922629, per second: 2069, peak per second: 3266
     Bytes: 1376559252, per second: 3096940, peak per second: 4880051
     Flow records processed: 25764957, per second: 42564, peak per second: 98124
Allocation:
     Blocks allocated: 20862, per second: 31, peak per second: 72
     Blocks freed: 17161, per second: 40, peak per second: 202
     Blocks unavailable: 58786, per second: 652, peak per second: 1120
Files:
     Files created: 52, per second: 0, peak per second: 0
     Files exported: 42, per second: 0, peak per second: 0
     Files destroyed: 42, per second: 0, peak per second: 0
Throughput:
     Uncompressed bytes: 2592070401, per second: 7297307,
     peak per second: 8630023
     Compressed bytes: 659600068, per second: 1858458, peak per second: 2198471
Packet drops:
     No memory: 58786, 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: 585981447, per second: 1313320, peak per second: 4857798
     FTP files: 48, per second: 0, peak per second: 0
     FTP failure: 8
Export channel: 0
     Current server: Primary
     Primary server state: FTP error, Secondary server state: Not configured
Export channel: 1
     Current server: Primary
     Primary server state: OK, Secondary server state: Not configured
```


CHAPTER 12

Passive Flow Monitoring Commands

clear passive-monitoring statistics

Syntax clear passive-monitoring statistics (all | interface *interface-name*)

Release Information Command introduced in Junos OS Release 7.6.

Description (M40e, M160, and M320 Series routers and T Series routers only) Clear statistics for one passive monitoring interface or for all passive monitoring interfaces.

Options **all**—Clear statistics for all configured passive monitoring interfaces.

interface *interface-name*—Clear statistics for the specified passive monitoring interface (**mo-*fpc/pic/port***).

Required Privilege Level network

List of Sample Output clear passive-monitoring statistics on page 236

Output Fields When you enter this command, you are not provided feedback on the status of your request.

Sample Output

clear passive-monitoring statistics

```
user@host> clear passive-monitoring statistics interface mo-5/0/0
```


show passive-monitoring error

Syntax show passive-monitoring error (* | all | mo-*fpc/pic/port*)

Release Information Command introduced before Junos OS Release 7.4.

Description (M40e, M160, and M320 Series routers and T Series routers only) Display passive monitoring error statistics.

Options *** | all | mo-*fpc/pic/port***—Display error statistics for monitoring interfaces. Use a wildcard character, specify all interfaces, or provide a specific interface name.

Required Privilege Level view

List of Sample Output show passive-monitoring error all on page 238

Output Fields Table 51 on page 237 lists the output fields for the **show passive-monitoring error** command. Output fields are listed in the approximate order in which they appear.

Table 51: show passive-monitoring error Output Fields

Field Name	Field Description
Passive monitoring interface	Name of the passive monitoring interface.
Local interface index	Index counter of the local interface.
Interface state	State of the passive monitoring interface: • **Monitoring**—Specified interface is actively monitoring. • **Disabled**—Specified interface has been disabled from the CLI. • **Not monitoring**—The interface is operational, but not monitoring. This condition occurs when an interface first comes online, or when the interface is operational, but no logical unit has been configured under the physical interface. • **Unknown**—Unknown state. • **Error**—An error occurred during the process of determining the state of the interface.
Error information	
Packets dropped (no memory)	Number of packets dropped because of memory shortage.
Packets dropped (not IP)	Number of non-IP packets dropped.
Packets dropped (not IPv4)	Number of packets dropped because they failed the IPv4 version check.
Packets dropped (header too small)	Number of packets dropped because the packet length or IP header length was too small.

Table 51: show passive-monitoring error Output Fields *(continued)*

Field Name	Field Description
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 free failures.
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 128 KB are being created per second.
Memory warning	Whether 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	Whether the memory has been overloaded. The response can be **Yes** or **No.**
PPS overload	Whether the PIC is receiving more packets per second than the configured threshold. The response can be **Yes** or **No**.
BPS overload	Whether the PIC is receiving more bits per second than the configured threshold. The response can be **Yes** or **No**.

Sample Output

show passive-monitoring error all

```
user@host> show passive-monitoring error all
Passive monitoring interface: mo-4/0/0, Local interface index: 44
Interface state: Monitoring
  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
Interface state: Not monitoring
  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
```

show passive-monitoring flow

Syntax show passive-monitoring flow (* | all | mo-*fpc/pic/port*)

Release Information Command introduced before Junos OS Release 7.4.

Description (M40e, M160, and M320 Series routers and T Series routers only) Display passive flow statistics.

Options *** | all | mo-*fpc/pic/port***—Display passive flow statistics for monitoring interfaces. Use a wildcard character, specify all interfaces, or provide a specific interface name.

Required Privilege Level view

List of Sample Output show passive-monitoring flow all on page 240

Output Fields Table 52 on page 239 lists the output fields for the **show passive-monitoring flow** command. Output fields are listed in the approximate order in which they appear.

Table 52: show passive-monitoring flow Output Fields

Field Name	Field Description
Passive monitoring interface	Name of the passive monitoring interface.
Local interface index	Index counter of the local interface.
Interface state	State of the passive monitoring interface: • **Monitoring**—Specified interface is actively monitoring. • **Disabled**—Specified interface has been disabled from the CLI. • **Not monitoring**—The interface is operational, but not monitoring. This condition occurs when an interface first comes online, or when the interface is operational, but no logical unit has been configured under the physical interface. • **Unknown**—Unknown state. • **Error**—An error occurred during the process of determining the state of the interface.
Flow information	
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.

Table 52: show passive-monitoring flow Output Fields *(continued)*

Field Name	Field Description
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 cflowd packets exported by an operational PIC.
Flows inactive timed out	Total number of flows that are exported because of inactivity.
Flows active timed out	Total number of long-lived flows that are exported because of an active timeout.

Sample Output

show passive-monitoring flow all

```
user@host> show passive-monitoring flow all
Passive monitoring interface: mo-4/0/0, Local interface index: 44
Interface state: Monitoring
  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
Interface state: Monitoring
  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
```

show passive-monitoring memory

Syntax show passive-monitoring memory (* | all | mo-*fpc/pic/port*)

Release Information Command introduced before Junos OS Release 7.4.

Description (M40e, M160, and M320 Series routers and T Series routers only) Display passive monitoring memory and flow record statistics

Options *** | all | mo-*fpc/pic/port***—Display memory and flow record statistics for monitoring interfaces. Use a wildcard character, specify all interfaces, or provide a specific interface name.

Required Privilege Level view

List of Sample Output show passive-monitoring memory all on page 241

Output Fields Table 53 on page 241 lists the output fields for the **show passive-monitoring memory** command. Output fields are listed in the approximate order in which they appear.

Table 53: show passive-monitoring memory Output Fields

Field Name	Field Description
Passive monitoring interface	Name of the passive monitoring interface.
Local interface index	Index counter of the local interface.
Memory utilization	
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 memory free	Total memory currently used and total amount of memory currently free (in bytes).

Sample Output

show passive-monitoring memory all

```
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
```


show passive-monitoring status

Syntax show passive-monitoring status (*| all | mo-*fpc/pic/port*)

Release Information Command introduced before Junos OS Release 7.4.

Description (M40e, M160, and M320 Series routers and T Series routers only) Display passive monitoring status.

Options *** | all | mo-*fpc/pic/port***—Display status for monitoring interfaces. Use a wildcard character, specify all interfaces, or provide a specific interface name.

Required Privilege Level view

List of Sample Output show passive-monitoring status all on page 244

Output Fields Table 54 on page 243 lists the output fields for the **show passive-monitoring status** command. Output fields are listed in the approximate order in which they appear.

Table 54: show passive-monitoring status Output Fields

Output Field	Output Field Description
Passive monitoring interface	Name of the passive monitoring interface.
Local interface index	Index counter of the local interface.
Interface state	Monitoring state of the passive monitoring interface. • **Monitoring**—PIC is actively monitoring. • **Disabled**—PIC has been disabled using the CLI. • **Not monitoring**—PIC is operational, but not monitoring. This condition can happen while the PIC is coming online, or when the PIC is operational but has no logical unit configured under the physical interface. • **Unknown**
Group index	Integer that represents the monitoring group of which the PIC is a member. **Group index** is a mapping from the group name to an index. It is not related to the number of monitoring groups.
Export interval	Configured export interval for cflowd records, in seconds.
Export format	Configured export format (only cflowd version 5 is supported).
Protocol	Protocol the PIC is configured to monitor (only IPv4 is supported).
Engine type	Configured engine type that is inserted in output cflowd packets.
Engine ID	Configured engine ID that is inserted in output cflowd packets.

Sample Output

show passive-monitoring status all

```
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

Passive monitoring interface: mo-4/1/0, Local interface index: 45
Interface state: Disabled

Passive monitoring interface: mo-4/2/0, Local interface index: 46
Interface state: Not monitoring
```

show passive-monitoring usage

Syntax show passive-monitoring usage (* | all | mo-*fpc/pic/port*)

Release Information Command introduced before Junos OS Release 7.4.

Description (M40e, M160, and M320 Series routers and T Series routers only) Display passive monitoring usage statistics.

Options *** | all | mo-*fpc/pic/port***—Display usage statistics for monitoring interfaces. Use a wildcard character, specify all interfaces, or provide a specific interface name.

Required Privilege Level view

List of Sample Output show passive-monitoring usage all on page 245

Output Fields Table 55 on page 245 lists the output fields for the **show passive-monitoring usage** command. Output fields are listed in the approximate order in which they appear.

Table 55: show passive-monitoring usage Output Fields

Output Field	Output Field Description
Passive monitoring interface	Name of the passive monitoring interface.
Local interface index	Index counter of the local interface.
CPU utilization	
Uptime	Time, in milliseconds, that the PIC has been operational.
Interrupt time	Total time that the PIC has spent processing packets since the last PIC reset.
Load (5 second)	CPU load on the PIC, averaged more than 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 more than 1 minute. The number is a percentage obtained by dividing the time spent on active tasks by the total elapsed time.

Sample Output

show passive-monitoring usage all

```
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%
```

